SENTIMENT ANALYSIS USING NATURAL LANGUAGE PROCESSING (NLP) FOR WALMART QUARTERLY EARNINGS REPORTS FOR AFTER-HOURS TRADERS

By

Pa Praesicharoen

September 1, 2023

An applied research project submitted for partial fulfilment of the requirements for the degree of Master of Science in Business Analytics

Abstract

This study aims to develop a tool using Natural Language Processing (NLP) for analysing Walmart's quarterly earnings reports for after-hours traders. The research evaluated various models, TF-IDF, GloVE, RoBERTa, FinBERT, and ChatGPT, on a dataset spanning from Q1 2019 to Q1 2024, assessing their reliability, performance, and processing time. The results indicate that ChatGPT outperformed other models in terms of accuracy and F1 score but had unsatisfied processing time. Additionally, the research also involved fine-tuning the models using Few-Shot Learning and adjusting the temperature settings. Although fine-tuning improved performance, it increased processing time and sometimes lowered F1 scores. Lowering the temperature enhanced accuracy and F1 score, but at the cost of increased processing time. The study concluded that while fine-tuning and temperature adjustments improved model performance, it introduced trade-offs between accuracy, F1 score, and processing time that must be carefully considered for different applications. The study highlights the potential of NLP for after-hours trading sentiment analysis but also underscores the need for further optimisation to address processing time and real-time applicability limitations.

Contents

List of Figures and Table

1. Introduction

Financial markets are always active, and often, a single report, such as a quarterly earnings report, can significantly impact them. These reports, mandated by the Securities and Exchange Commission (SEC) using Form 10-Q (J87as, 2021), act as a company's health check-up and can influence stock prices. For example, even if a company reports unsatisfactory revenue, its stock price might still rise if the market has low expectations or if the future outlook is positive (Kuepper, 2019).

Interestingly, about 95% of public companies announce their earnings outside the regular trading hours of 9 a.m. to 4 p.m. Eastern time ("Traders are surprisingly slow to respond to off-hours earnings announcements," 2019). This has led to the rise of after-hours trading, which occurs between 4 p.m. and 8 p.m. EST and is facilitated by electronic communication networks (ECNs) (Jones, 2023). This allows investors to react quickly to significant news, such as earnings updates or major company announcements, and adjust their strategies accordingly (Levy, 2023; Chen, 2023). However, investors must be prepared to make quick decisions as a report could cause a stock's price to fluctuate significantly (Michael, 2023).

Advanced tools, specifically Natural Language Processing (NLP), play a crucial role. NLP, a combination of artificial intelligence and linguistics, enables computers to understand human language as we do (Khurana et al., 2022). With the help of Large Language Models (LLMs), it can quickly extract core insights from vast financial documents. This includes diverse sources like news articles and detailed financial reports, facilitating sentiment analysis, topic recognition, and more (Masłowska and Netguru, 2023).

Walmart was chosen for this sentiment analysis project due to its global presence and the vast collection of reports available. The goal is to use NLP to develop a specialized tool for sentiment analysing Walmart's quarterly earnings reports, particularly for after-hours trading. Existing tools may not capture all details in such comprehensive reports, so this project will explore whether NLP and LLMs can help transform complex financial narratives into easily understandable pointers for traders.

2. Literature Review

Financial sentiment analysis evolved significantly with Loughran and McDonald (2014) pioneering the sentiment metric by categorizing words in 10-K reports (1994-2012) into optimistic or pessimistic using the Diction dictionary. However, 10-Q filings brought extraction challenges due to diversity, volume, and non-machine-readable formats. Zhang et al. (2021) addressed this using rule-based methods and Convolutional Neural Networks (CNNs) tailored for image classification to extract data from textual sources, highlighting typographical nuances.

Yet, the field of Natural Language Processing (NLP) comes with its set of challenges. As described by Khurana et al. (2022), the ever-changing nature of language can be tricky for machines to understand. This includes shifting meanings of words based on context and managing words that sound alike but mean different things. On top of this, words that mean one thing in a field like education might have a completely different meaning in healthcare, making the process even more intricate.

In this context, Large Language Models (LLMs) stood out as game-changers. These advanced text generation tools, backed by major players like OpenAI and Microsoft, received significant development and tuning. Lipenkova (2022) traced their progression, noting the remarkable increase in size from the days of BERT to newer models like Megatron-Turing NLG.

However, challenges remain. The success of NLP is closely linked to the quality of its input. For many, the details of NLP can be puzzling, and there's noticeable untapped potential in models, particularly those from OpenAI.

In response, our project steps in to address these gaps. By leveraging the strengths of GPT 3.5, we aim to push forward in the area of sentiment analysis. By allowing users to add their own keywords and financial updates to earnings report reviews, we hope to provide a customized and thorough sentiment perspective for investors, marking a fresh step forward in financial sentiment analysis.

3. Exploratory Data Analysis (EDA)

For Walmart's quarterly reports, the first page consistently contains key highlights and crucial data, followed by two pages of financial information in tables. Interestingly, Figure 1 shows that the fourth page typically has the most text, but it is mostly a repetitive "About Walmart", and "Forward-Looking Statement" section, which lacks relevant information.

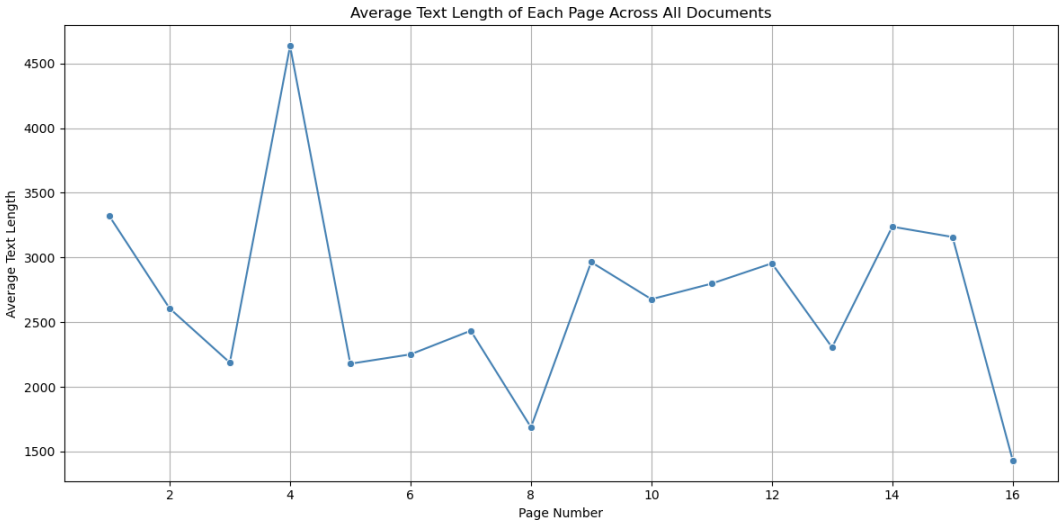


Figure 1 – Average text length of each page across all documents

The fourth-quarter report is longer than others due to the inclusion of "Full-year highlights," as shown in Figure 2.

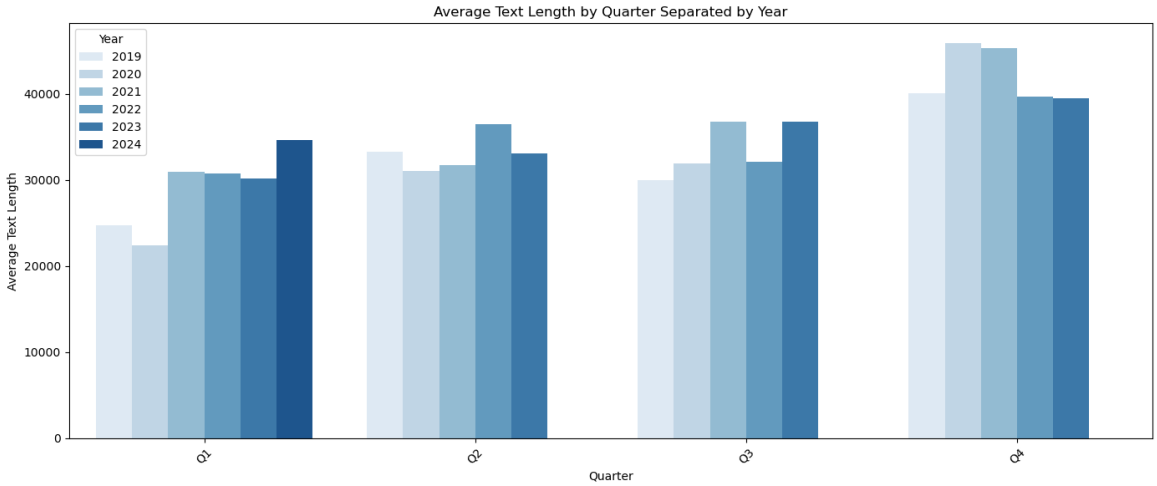


Figure 2 – Average text length by quarter separated by year

Figure 3 shows the quarterly reports' sentiment distribution grouped by year, revealing the highest distribution of positive sentiment, followed by neutral and the least negative.

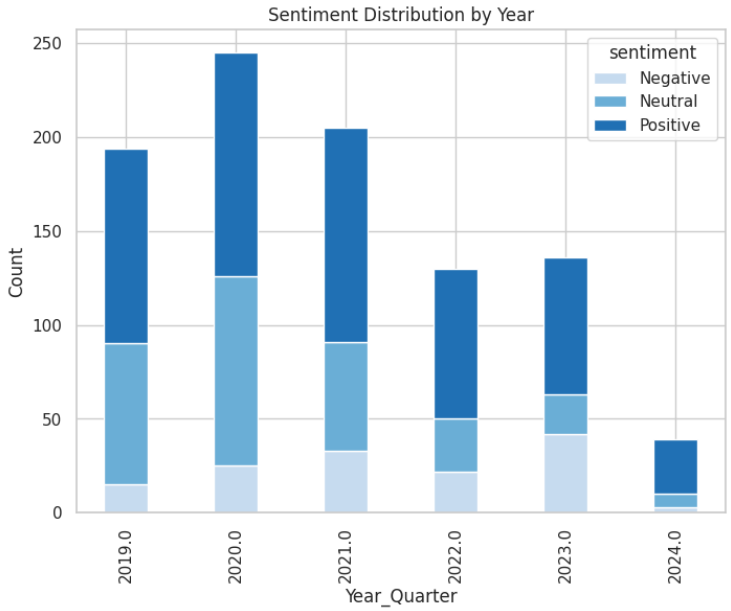


Figure 3 – Sentiment distribution by year

Figure 4 reveals a trend of increasing word counts in reports since 2019, peaking in 2021 due to the addition of an "Additional Highlights" section, illustrating the ongoing changes in report formats.

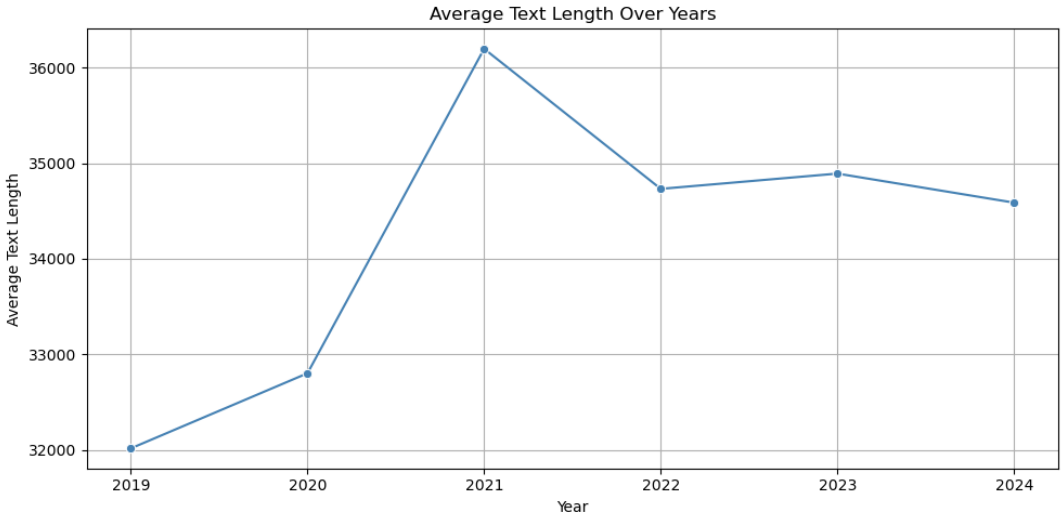


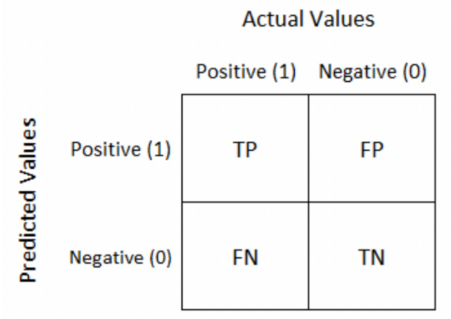
Figure 4 – Average text length over years

4. Methodology

4.1 Evaluation Measures

We assessed our after-hours trading sentiment analysis using accuracy, F1 score, and processing time. Higher accuracy and F1 scores signify better performance, whereas lower processing time guarantees prompt insights. The confusion matrix helps determine TP, TN, FP, and FN for each sentiment (positive, neutral, negative).

Sentiment	TP Definition	FP Definition	TN Definition	FN Definition
Positive	Correctly identified as positive	Incorrectly labeled as positive	Rightly marked as not positive	Misclassified positives as neutral or negative
Neutral	Correctly identified as neutral	Incorrectly labeled as neutral	Rightly marked as not neutral	Misclassified neutrals as positive or negative
Negative	Correctly identified as negative	Incorrectly labeled as negative	Rightly marked as not negative	Misclassified negatives as neutral or positive



Accuracy

Accuracy is the ratio of correct classifications to total test cases (Ghosh, 2023), e.g., 90% correct prediction means 90% accuracy (Korstanje, 2022).

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

F1 Score

The F1 score, a combination of precision and recall, is most useful when false positives and negatives carry similar costs, additional data has minimal impact, and true negatives are plentiful (Agrawal, 2023). We utilized the weighted F1 score to account for class imbalances and weigh scores based on class instances (Leung, 2022).

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

$$\text{Recall} = \frac{|TP|}{|TP| + |FN|}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Processing Time

Beyond accuracy, processing time is crucial as it determines the model's practical utility in real-world applications.

```
start_time = time.time() # Record start time

end_time = time.time()  # Record end time
Processing Time = end_time - start_time
```

4.2 Part I – NLP Models Comparison

4.2.1 Data Curation

We created a labelled sentiment dataset using Q1 2019 – Q1 2024 Walmart quarterly reports. Two individuals classified sentiments (positive, neutral, negative) at bullet-point and paragraph levels, based on the report format. Recurrent neutral sections with insignificant sentiment value, such as "About Walmart" and "Forward-Looking Statement", were removed. The sentiments labelled by both individuals were consolidated for consistency.

Year	Quarter	Text	Pa Sentiment	V Sentiment	Conclude Sentiment	Section
2024	Q1	Strong revenue growth of 7.6%; operating income growing faster at 17.3%	Positive	Positive	Positive	Headline
2024	Q1	Guides Q2 and raises FY24 outlook	Neutral	Neutral	Neutral	Headline
2024	Q1	Consolidated revenue of $152.3 billion, up 7.6%, or 7.7% in constant currency ("cc")1	Positive	Positive	Positive	First Quarter Highlights
2024	Q1	Consolidated gross margin rate declines 18 basis points on mix of sales	Negative	Negative	Negative	First Quarter Highlights

Figure 5 – Example of a labelled sentiment dataset

Limitations included occasional discrepancies in sentiment labelling, particularly in borderline areas, and the analysis was limited to twenty reports over five years due to the time-intensive nature of the task.

4.2.2 Data Preprocessing

We divided the labelled sentiment dataset into a Training set (Q1 2019 - Q1 2023 reports) and a Testing set (Q2 2023 - Q1 2024 reports). Missing 'text' column values, which occurred due to the Excel to CSV conversion process, were addressed by removal.

A raw version was used for RoBERTa, FinBERT, and ChatGPT models, which can handle unprocessed content. For TF-IDF and GloVE models, the data underwent extensive processing, including HTML tag removal, lowercase conversion, abbreviation expansion, tokenization, punctuation and numeric removal, part-of-speech tagging, and lemmatization, to optimise model performance.

```
df_cleaned = df.dropna(subset=['text'])
df_cleaned['text'].fillna('No Text', inplace=True)
df_cleaned['text'] = df_cleaned['text'].astype(str)
df_1 = df_cleaned
df_cleaned.head()
```

Figure 6 – Python code for preprocessing labelled sentiment dataset 1

```
def remove_tags(txt):
    txt = re.sub(re.compile('.<.*?>'), '', txt)
    txt = txt.translate(str.maketrans("", "", string.punctuation))
    # lower text
    txt = txt.lower()
    # expand shoten words
    txt = decontract(txt)
    txt = txt.strip()
    txt = txt.translate(str.maketrans("", "", string.punctuation))
    # tokenize text and remove puncutation
    txt = [word.strip(string.punctuation) for word in txt.split(" ")]
    # remove words that contain numbers
    txt = [word for word in txt if not any(c.isdigit() for c in word)]
    # remove stop words
    stop = stopwords.words('english')
    txt = [x for x in txt if x not in stop]
    # remove empty tokens
    txt = [t for t in txt if len(t) > 0]
    # pos tag text
    pos_tags = pos_tag(txt)
    # lemmatize text
    txt = [WordNetLemmatizer().lemmatize(t[0], get_wordnet_pos(t[1])) for t in pos_tags]
    # join all
    txt = " ".join(txt)
    # remove numbers and specific signs
    txt = re.sub(r'\d+', '', txt)
    txt = re.sub(r'\W+', ' ', txt)
    return txt
```

Figure 7 – Python code for preprocessing labelled sentiment dataset 2

4.2.3 Training, Testing and Evaluation of Models Performance

We examined the reliability and performance of five models: TF-IDF, GloVE, RoBERTa, FinBERT, and ChatGPT for sentiment analysis of Walmart's earnings reports. TF-IDF, GloVE, and RoBERTa were trained on a Train dataset (Q1 2019 - Q1 2023 reports), which allowed them to discern patterns and form predictions anchored on prior learnings when faced with new data from the Test dataset (Q2 2023 - Q1 2024 reports). FinBERT and ChatGPT were tested without training before. The practical evaluation involved accuracy, F1 score, and processing time tested quarterly, mirroring real-world use.

4.2.4 Model Justification

TF-IDF (Countvectorizer + Multinomial Naive Bayes)

This model leans on CountVectorizer for term frequencies, balancing it with document inverse frequency—a measure that assesses word relevance across documents (Gulati, 2022). Given its efficacy in text classification and topic modelling, we paired it with the multinomial naïve Bayes, known for its efficiency in document categorisation based on content analysis (Sriram, 2022).

GloVE (Word2Vec + Logistic Regression)

GloVE's primary strength lies in its ability to encapsulate the meanings of words by evaluating the comprehensive structure of an entire corpus. It delves into understanding the intrinsic relationships among words based on global statistics. While many unsupervised algorithms traditionally lean on word frequencies and co-occurrence counts (Chakravarthy, 2021), GloVE stands out. Its widespread acclaim in NLP-based sentiment analysis and its documented superiority over Word2Vec make it an invaluable tool.

RoBERTa

A transformation of the BERT model, RoBERTa thrives on its self-attention mechanism, offering contextualized word interpretations. With adjustments like dynamic masking and larger batch-training sizes (GeeksforGeeks, 2023), it presents as a heavyweight contender for ChatGPT in our lineup. However, a caveat is its variable training processing time.

FinBERT

An offshoot of Google's BERT is specifically designed for financial sentiment analysis (W&B, 2023). It was meticulously trained on diverse financial datasets, including corporate filings from the SEC's EDGAR website, analyst reports from the Thomson Investext database, and transcripts from SeekingAlpha, spanning years from 1994 to 2019 (Huang, Wang, and Yang,

2020). Further refined using 10,000 manually annotated sentences from analyst reports, this model, known as FinBERT-tone, excels in financial tone analysis (yiyanghkust/finbert-tone · Hugging Face). Consequently, we saw no need for additional training or fine-tuning.

ChatGPT

OpenAI's GPT-3, a leading language generation model (Brown, 2020) with 175 billion parameters (Leippold, 2023). The GPT3.5-turbo-16k model was selected due to its cost-efficiency coupled with its high-level performance. Additionally, the GPT-3.5-turbo-16k provides four times the context length compared to the GPT-3.5-turbo, allowing it to handle approximately 20 pages of text in one request, as pointed out by Greyling (2023). A significant perk of ChatGPT is its no-training feature, enabling immediate deployment with top accuracy and F1 scores. Notably, text segmentation into multiple chunks due to token length restrictions, prompt sensitivity affecting accuracy, and the need for consistent output stabilisation using a temperature setting.

```python
def chunks(lst, n):
    """Yield successive n-sized chunks from lst."""
    for i in range(0, len(lst), n):
        yield lst[i:i + n]

def generate_prompt(sentences_chunk):
    sentences = "\n".join(f"- text: {text}" for text in sentences_chunk)
    prompt = f"""
    Please analyze the financial sentiments of the statements.
    Please Identify the change or sentiment from the difference between GAAP EPS and Adjusted EPS.
    If the adjusted one is higher, you can assume it's positive, or otherwise.
    The affirmative sentences are considered neutral sentiments.
    {sentences}

    For each row:
    1. Classify the sentiment as 'positive', 'negative', or 'neutral'.

    Format the answer as:
    Text:::Sentiment
    Text:::Sentiment
    ...
    """
    return prompt


# Estimate the number of sentences per chunk based on the token limit.
MAX_SENTENCES_PER_CHUNK = 20  # Adjust this value based on the average length of your sentences

all_results = []
for chunk in chunks(gpt_2024_q1['text'].tolist(), MAX_SENTENCES_PER_CHUNK):
    print(chunk)
    prompt = generate_prompt(chunk)

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo-16k",
        temperature=0.2,
        messages=[
            {"role": "system", "content":
                """You are a stock and financial sentiment analysis model specialized in
                Retail businesses and always up-to-date with the latest news.
                You have the capability to analyze the sentiment of Walmart's earnings release
                and classify it as positive, negative, or neutral.
                """},
            {"role": "user", "content": prompt}
        ]
    )

    model_21_output_df = output_to_df_gpt35(response)
    all_results.append(model_21_output_df)
    time.sleep(3)  # To handle rate limits
```

Figure 8 – Python code showing text segmentation and prompts used in ChatGPT model

ChatGPT emerges as the superior model based on the highest F1 score, and we are dedicated to enhancing it further, particularly in optimising its runtime.

4.3 Part II - Keyword Sentiment Analysis Using ChatGPT 3.5 Turbo 16k Model

4.3.1 Data Gathering

Automating Extraction of Walmart's Quarterly Reports

We automated the extraction of forty Walmart's quarterly earnings reports from the past decade using Python, specifically the Selenium, PyPDF2, Fitz, pdfplumber, pandas, and Numpy libraries. The process involved the following steps:

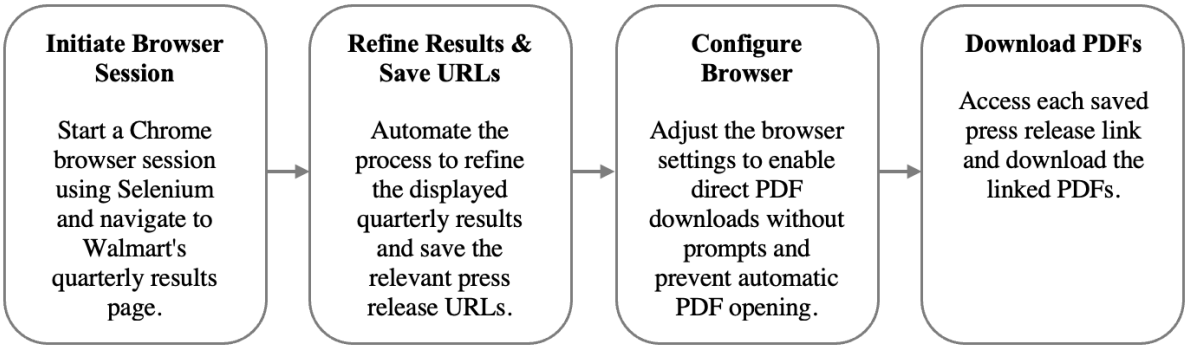


Figure 9 – Process of automating the extraction of Walmart's quarterly reports

Scraping Financial Times for Relevant News Articles

We retrieved Walmart news articles from the Financial Times over the last decade using a process similar to Walmart's quarterly reports scraping. The process involved the following steps:

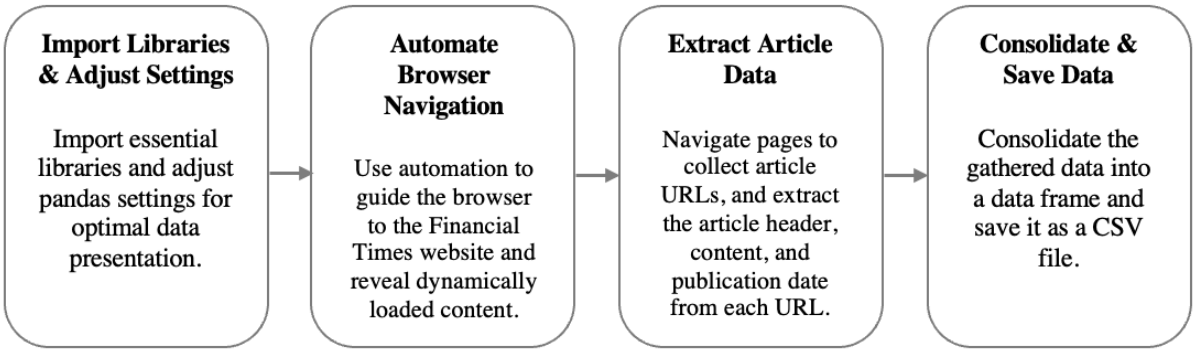


Figure 10 – Process of scraping Financial Times for relevant news articles

Constructing a Financial Keywords Repository

We collected key financial terms from twenty Walmart quarterly reports, including 'revenue', 'sales', 'EPS', 'ROA', etc., to systematically search each report in the next research stages.

Establishing The Labelled Keywords Dataset

Two people independently labelled keywords, discussing finalising sentiment labels, ensuring consistency, and minimizing biases. We selected keywords with clear sentiment, excluding ambiguous terms, and evaluated non-financial keywords impacting Walmart's image or operations, considering their contextual sentiment within the reports.

This dataset is foundational for our analysis, enabling models, especially ChatGPT and its variants, to effectively recognize keywords and their associated sentiments. It serves as a benchmark for a more informed assessment of ChatGPT's predictive capabilities.

However, challenges include potential human biases in sentiment labelling and the extensive time required for manual curations.

Year	Quarter	Keywords	Type	Pa Sentiment	V Sentiment	Conclude Sentiment	Remarks
2024	Q1	Advertising	non-financial	Positive	Positive	Positive	
2024	Q1	Revenue	financial	Positive	Positive	Positive	
2024	Q1	Walmart U.S Comp Sales	financial	Positive	Positive	Positive	
2024	Q1	Sam's Club Comp Sales	financial	Positive	Positive	Positive	
2024	Q1	Walmart International Net Sales	financial	Positive	Positive	Positive	
2024	Q1	Consolidated Operating Income	financial	Positive	Positive	Positive	
2024	Q1	Consolidated Operating Expenses	financial	Positive	Positive	Positive	
2024	Q1	Consolidated Gross Margin Rate	financial	Negative	Negative	Negative	
2024	Q1	Adjusted EPS	financial	Positive	Positive	Positive	
2024	Q1	Free Cash Flow	financial	Positive	Positive	Positive	
2024	Q1	ROI	financial	Negative	Negative	Negative	
2024	Q1	ROA	financial	Negative	Negative	Negative	
2024	Q1	Profit	financial	Neutral	Positive	Positive	
2024	Q1	Total Debt	financial	Neutral	Neutral	Neutral	
2024	Q1	eCommerce	non-financial	Positive	Positive	Positive	
2024	Q1	Inventory	financial	Neutral	Neutral	Neutral	
2024	Q1	Market Share	non-financial	Positive	Positive	Positive	
2024	Q1	Membership Income	financial	Positive	Positive	Positive	
2024	Q1	PhonePe	non-financial	Positive	Positive	Positive	
2024	Q1	Currency	financial	Neutral	Positive	Neutral	did not mention any effects from currency flucuation
2024	Q1	Repurchase	financial	Positive	Positive	Positive	

Figure 11 – Example of labelled keywords dataset

4.2.2 Data Preprocessing

Preprocess Walmart's Quarterly Reports PDFs

We designated a directory, pdf_folder, for the PDFs and initialised an all_text_data list for extracted text. Using fitz.open(), we extracted and standardized text from each file and stored it in the all_text_data list. This data was structured into a pandas dataframe, df, ready for analysis.

```
# Get a list of all PDF files in the folder
pdf_files = [file for file in os.listdir(pdf_folder) if file.lower().endswith('.pdf')]

all_text_data = []  # Initialize a list to hold the extracted text data

for pdf_file in pdf_files:
    pdf_path = os.path.join(pdf_folder, pdf_file)

    doc = fitz.open(pdf_path)

    page_texts = []  # Initialize a list to hold page texts for this file

    for i, page in enumerate(doc):
        page_text = page.get_text("text")
        page_text = page_text.replace('', ' ').lower()
        page_texts.append(page_text)  # Add page text to the list

    all_text_data.append((pdf_file, page_texts))  # Add filename and list of page texts to the list

    doc.close()  # Close the PDF after processing

# Create a DataFrame from the text data
columns = ["file_name", "page_text"]
df = pd.DataFrame(all_text_data, columns=columns)

# Display the DataFrame
print(df)
```

Figure 12 – Python code for preprocessing Walmart's Quarterly Reports PDFs 1

Challenges included unclear table formatting and format changes in 2024. To address this, we employed specialized functions from the PyMuPDF package to enhance focus and ensure no vital data was missed. Irrelevant sections were filtered out, and ChatGPT's output was transformed into a dataframe.

```python
# Extract page that starts with a keyword list
def page_starts_with(text_list, keyword_list):
    matching_items = []
    for item in text_list:
        item_str = str(item)  # Convert the element to a string
        for keyword in keyword_list:
            if item_str.lower().strip().startswith(keyword.lower()):
                matching_items.append(item)
                break
    return matching_items


def extract_sections_containing_word(sentences, words):
    # Convert sentences to lowercase for case-insensitive matching
    lower_sentences = [sentence.lower() for sentence in sentences]

    # Convert words to lowercase
    lower_words = []
    for word in words:
        if isinstance(word, list):
            lower_words.extend([w.lower() for w in word])
        else:
            lower_words.append(word.lower())

    # Filter the sentences to keep only those that contain any of the words
    sections_with_word = [sentence for sentence in lower_sentences if any(word in sentence for word in lower_words)]

    return sections_with_word


# Remove forward-looking statements
def remove_forward_looking_statements(text_list):
    cleaned_list = [item for item in text_list if not str(item).lower().strip().startswith('forward-looking statements')]
    return cleaned_list


# Convert chatgpt3.5 output to dataframe
def output_to_df_gpt35(response):
    sentiment_yes_no_neu = response['choices'][0]['message']['content'].strip()
    sentiment_results = sentiment_yes_no_neu.split("\n")
    # sentiment_results = sentiment_yes_no_neu.split(":::")
    filtered_results = [result for result in sentiment_results if result.strip()]
    filtered_results
    final_df = pd.DataFrame([result.split(":::") for result in filtered_results], columns=["Keyword", "Sentiment", "Reason"])
    # Lowercase every column
    final_df = final_df.applymap(lambda x: x.lower() if isinstance(x, str) else x)
    final_df['Keyword'] = final_df['Keyword'].str.strip()
    return final_df
```

Figure 13 – Python code for preprocessing Walmart's Quarterly Reports PDFs 2

The script also handled data's variable structure across PDF pages. The model_1_preprocessing() function targeted the first report page while model_2_preprocessing() processed pages 1-4, extracting only sentences with relevant keywords to maintain efficiency.

```python
# For Training dataset Earlier than 2024
def model_1_preprocessing(text):
    # page_1_text = text[0]
    page_1_text_split = re.split('\s{3,}|•', text[0])
    cleaned_page_1_text_split = [text.replace('\n', '').lower() for text in page_1_text_split]
    return cleaned_page_1_text_split

def model_2_preprocessing(text):
    page_2_to_4_text = text[1:4]
    cleaned_page_2_to_4_text = [re.sub(r'\n“[^”]*”\n', '', text) for text in page_2_to_4_text]
    # cleaned_page_1_text_list = [cleaned_page_1_text]
    cleaned_page_1_text_list = model_1_preprocessing(text)
    cleaned_page_1_text_list.extend(cleaned_page_2_to_4_text)
    cleaned_page_1_to_4_text =  cleaned_page_1_text_list + cleaned_page_2_to_4_text
    # cleaned_page_1_to_4_text
    other_important_text = page_starts_with(text, ['business highlights', 'free cash flow', 'adjusted eps', 'return on investment'])
    cleaned_other_important_text = [re.sub(r'\n“[^”]*”\n', '', text) for text in other_important_text]
    # cleaned_other_important_text
    model_2_text = cleaned_other_important_text + cleaned_page_1_to_4_text
    model_2_text = remove_forward_looking_statements(model_2_text)
    cleaned_model_2_text = [text.replace('\n', '') for text in model_2_text]
    return cleaned_model_2_text
```

Figure 14 – Python code for preprocessing Walmart's Quarterly Reports PDFs 3

Preprocessing Financial Times for Relevant News Articles

The article_date column is segmented into Month, Day, and Year. A Quarter column is introduced to match Walmart's fiscal year, incrementing the Year column by 1. In the df_ft_news_grouped dataframe, newline characters in article_body are replaced with spaces and text is lowercased.

```python
df_ft_news[["Month", "Day", "Year"]] = df_ft_news["article_date"].str.split(expand=True)
df_ft_news["Quarter"] = (df_ft_news["Month"].str.lower().replace({
    "january": 'Q4', "february": 'Q1', "march": 'Q1',
    "april": 'Q1', "may": 'Q2', "june": 'Q2',
    "july": 'Q2', "august": 'Q3', "september": 'Q3',
    "october": 'Q3', "november": 'Q4', "december": 'Q4'
}))
df_ft_news["Year"] = df_ft_news["Year"].astype(int) + 1
df_ft_news.head(3)
```

Figure 15 – Python code for preprocessing Financial Times

	Unnamed: 0	article_date	article_header	article_body	Month	Day	Year	Quarter
0	https://www.ft.com/content/8abcbadb-72e1-4efe-9ba3-dc18870a94c5	MAY 20 2023	Amazon falls behind Walmart in battle for India's online shoppers	Walmart is beating Amazon in the battle for online consumers in India, a rare ecommerce win for ...	MAY	20	2024	Q2
1	https://www.ft.com/content/e46f6a2c-68b0-4554-aaa8-b7525e6f8f3d	OCTOBER 18 2022	Book review: Still Broke by Rick Wartzman	By the time Sam Walton opened his first Wal-mart in Rogers, Arkansas, in 1962, his playbook to b...	OCTOBER	18	2023	Q3
2	https://www.ft.com/content/bf0ea662-bdaf-4445-a142-0a41e6308d83	JULY 27 2022	A new CEO for Credit Suisse	Your browser does not support playing this file but you can still download the MP3 file to play ...	JULY	27	2023	Q2

Figure 16 – Example of preprocessed Financial Times for relevant news articles

Preprocess The Labelled Keyword Dataset

The labelled keyword dataset is loaded into a dataframe, discarding entries with missing 'Year' values, and standardizing the 'Year' column to integer format. The 'Keywords', 'Keywords_Type', and 'Conclude_Sentiment' columns are transformed to lowercase for consistency.

	Year	Quarter	Keyword	Keyword_Type	Conclude_Sentiment
0	2024	Q1	advertising	non-financial	positive
1	2024	Q1	revenue	financial	positive
2	2024	Q1	walmart u.s comp sales	financial	positive
3	2024	Q1	sam's club comp sales	financial	positive
4	2024	Q1	walmart international net sales	financial	positive

Figure 17 – Example of preprocessed labelled keyword dataset

Prepare Final Dataset for Model 1

The preprocessed keywords dataset was narrowed to include only financial keywords. The 'Keywords' column is transformed to lowercase, and data is grouped by Year, Quarter, file_name, and Type. Then, Walmart's Quarterly Reports and pre-processed keywords dataset for model 1 were merged based on the file_name column.

file_name	page_text	Year	Quarter	Type	Keywords
2Q22-PR.pdf	[walmart u.s. q2 comp sales1 grew 5.2%; 14.5% two-year stack; comp transactions strong at 6.1%\n...	2022	Q2	Financial	[revenue, sam's club comp sales, membership income, walmart international net sales, consolidate...
4Q20-PR.pdf	[nyse: wmt\nfebruary 18, 2020\nstock.walmart.com\nwalmart u.s. q4 comp sales1 grew 1.9% and walm...	2020	Q4	Financial	[revenue, walmart u.s comp sales, sam's club comp sales, walmart international net sales, operat...
4Q22-PR.pdf	[walmart inc. net sales exceed $150 billion in q4\nwalmart u.s. net sales exceed $105 billion in...	2022	Q4	Financial	[revenue, walmart u.s sales, consolidated gross profit rate, sam's club comp sales, membership i...

Figure 18 – Example of the final dataset for model 1 analysis

Prepare Final Dataset for Models 2 and 3

The data for Models 2 and 3's analysis utilised both financial and non-financial keywords. All entries in the 'Keywords' column were standardized to lowercase. The dataset was then structured by grouping keywords based on their Year, Quarter, and file_name. The Quarterly Reports and keyword dataset were merged using the file_name column. Next, the 'Year' columns in both this dataset and the news article dataset are standardized to integer type. Lastly, the two datasets are merged on 'Year' and 'Quarter' columns, resulting in the final dataset for models 2 and 3 analysis.

file_name	page_text	Year	Quarter	Keywords	article_body
2Q22-PR.pdf	[walmart u.s. q2 comp sales1 grew 5.2%; 14.5% two-year stack; comp transactions strong at 6.1%\n...	2022	Q2	[revenue, market share, advertising, walmart u.s comp transactions, sam's club comp sales, ecomm...	[retailers offered a rosy picture on the health of the us consumer on tuesday after reporting st...
4Q20-PR.pdf	[nyse: wmt\nfebruary 18, 2020\nstock.walmart.com\nwalmart u.s. q4 comp sales1 grew 1.9% and walm...	2020	Q4	[revenue, walmart u.s comp sales, ecommerce, sam's club comp sales, walmart international net sa...	[synchrony financial was heading toward its best day ever on wall street after extending a credi...
4Q22-PR.pdf	[walmart inc. net sales exceed $150 billion in q4\nwalmart u.s. net sales exceed $105 billion in...	2022	Q4	[revenue, walmart u.s sales, market share, ecommerce, consolidated gross profit rate, sam's club...	[the supply chain problems of us retailers are being exacerbated by computer programs known as "...

Figure 19 – Example of the final dataset for models 2 and 3 analysis

4.3.3 Models Justification

Model 1

Focuses on summary sentiment analysis of key financial terms from Walmart's quarterly reports, like Total Revenue and Adjusted EPS. We customized our keyword list to include company-specific terms, enhancing the model's utility for investors.

Model 2

Provides summary sentiment analysis for predetermined and user-specified financial or non-financial keywords, acknowledging the limitations of existing methods like the Loughran and McDonald (2014) approach. This ensures a more nuanced and comprehensive analysis.

Model 3

Analyses predetermined and user-specified keywords from Models 1 and 2, and news excerpts from The Financial Times, accounting for social and global events. This addresses the limitations of dictionary-based methods by accounting for changes in language and sentiment expressions over time.

4.3.4 Variations of Models

Model	Description
1.1	Analyzes only the first page of the quarterly report, including main highlights.
1.2	Analyzes the first four pages and other relevant sections, including crucial financial metrics.
1.3	Extracts specific sentences containing financial keywords from the quarterly report, optimizing speed.
2.1	Similar to 1.1 but includes user-specified keywords.
2.2	Similar to 1.2 but includes user-specified keywords.
2.3	Similar to 1.3 but includes user-specified keywords.
3.1	Similar to 2.1 but also analyzes Financial Times excerpts.
3.2	Similar to 2.2 but also analyzes Financial Times excerpts.
3.3	Similar to 2.3 but also analyzes Financial Times excerpts.

Table 1 – Variations of models for Part II

4.3.5 Train and Test ChatGPT Models

Train Models

ChatGPT was used for automating model training, evaluation, and fine-tuning due to its high F1 score. Nine model variations (1.1-3.3) were trained and evaluated using a dataset from Q1 2019 to Q1 2023. Metrics used for evaluation included accuracy, F1 score, and processing time, tested quarterly. Despite observed instability in all models, 1.3, 2.3, and 3.3 showed the highest accuracy, F1 scores, and shortest runtime, and were selected for fine-tuning using 'few-shot learning' in ChatGPT.

Test Models 1.3, 2.3, and 3.3

These models were tested on a dataset from Q2 2023 to Q1 2024 to ensure robust performance despite potential changes in data format, using the same evaluation metrics to mirror real-world usage.

4.3.6 Finetune Using ChatGPT with Prompt Engineering (Few-Shot Learning) and Reducing Temperature

Fine-tuning ChatGPT using prompt engineering and few-shot learning is advantageous as it allows for the rapid development of a domain-specific and accurate model in a cost-effective manner. This method, applied to models 1.3, 2.3, and 3.3 with datasets from Q2 2023 to Q1 2024, enables ChatGPT to understand financial jargon, thereby enhancing its ability to extract and interpret relevant information.

Moreover, reducing the temperature from 0.3 to 0.2 results in more deterministic and focused predictions, whereas a value of 0.3 induces more randomness and diversity. However, limitations include potential overfitting due to limited data and challenges in designing effective prompts for complex tasks.

```
response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo-16k",
    temperature = 0.2,
    messages=[
        {"role": "system", "content":
            """You are a financial sentiment analysis model specializing in the retail business and you are helping after-hours
            traders to decide whether or not they should invest in Walmart's stock.

            You are up to date with the latest financial and global news.
            You will do this by analysing the sentiment of the text in Walmart's quarterly earnings release reports.
            You can classify each sentence in the text as positive, negative, or neutral.
            Please use concise and formal language when presenting results.

            Consider the entire keyword or parts of it in your analysis.
            You always have an answer for every keyword.
            Make sure you answer each and every keyword and do not miss any."""},

        # Example messages
        {"role": "system", "content": "you are provided with the following examples for few-shot learning:"},
        {"role": "user", "content": "adjusted eps:::positive:::the q3 fy23 gaap eps was reported at ($0.66), but the adjusted eps exc
        {"role": "user", "content": "international net sales:::positive::: walmart international net sales were $24.4 billion, an inc
        {"role": "user", "content": "consolidated net sales:::positive::: consolidated net sales growth is expected to be about 4.5%.
        {"role": "user", "content": "net cash:::negative::: net cash provided by operating activities was $9.2 billion for the six mo
        {"role": "user", "content": "consolidated operating expenses:::negative::: consolidated operating expenses as a percentage of
        {"role": "user", "content": "adjusted eps:::positive::: adjusted eps of $1.50 excludes the effects, net of tax, of $1.11 from
        {"role": "user", "content": "revenue:::positive:::total revenue was $141.0 billion, up 2.4%, negatively affected by approxima
        {"role": "user", "content": "free cash flow:::negative:::we generated free cash flow of $7.4 billion for the six months ended
        {"role": "user", "content": "capital expenditures:::negative:::we generated free cash flow of $7.4 billion for the six months
        {"role": "user", "content": "currency:::negative:::excluding currency, total revenue would have increased 0.6% to $138.6 bill

        # User message for analysis
        {"role": "user", "content": f"""
        Based on the provided earnings release text please analyze the sentiments of the specified keywords in the earnings release.
        - Earnings Release Text: {model_2_data}
        - Keywords: {keywords}

        For each keyword:
        1. Classify the sentiment as 'positive', 'negative', or 'neutral'.
        2. Provide a reason by behind the keyword sentiment output in the earnings release text.
        3. Make sure you answer each and every keyword.
        4. If keyword is “Adjusted EPS” compare the value associated with this keyword with the value for “GAAP EPS”.
        If “Adjusted EPS” value is higher than “GAAP EPS” value the sentiment is positive

        Lastly, give the 'Overall Sentiment' of the entire earnings release and justify your answer.

        Format the answer as:
        Keyword:::Sentiment:::Reason
        Keyword:::Sentiment:::Reason
        ...
        Overall Sentiment:::Sentiment:::Reason
        """}
```

Figure 20 – Python code showing set temperature = 0.2 and prompts for few-shot learning

5. Results Analysis

5.1 Part I – NLP Models Comparison

The objective of the project was to perform sentiment analysis on Walmart's quarterly earnings reports for after-hours traders using Natural Language Processing (NLP). We evaluated five distinguished models: TF-IDF, GloVE, RoBERTa, FinBERT, and ChatGPT based on their reliability, performance, and processing time. These models were trained using a dataset spanning Q1 2019 to Q1 2023 and tested on a dataset from Q2 2023 to Q1 2024. We evaluated the models based on three key metrics: accuracy, F1 score, and processing time.

The results of the evaluation are summarized in the tables below:

Accuracy Results of Testing All Models

Model	Q1 2024	Q4 2023	Q3 2023	Q2 2023	Average
TF-IDF	79.49%	75.86%	62.50%	55.56%	**68.35%**
GloVE	76.92%	75.86%	70.83%	58.33%	**70.49%**
RoBERTa	84.62%	82.76%	85.42%	66.67%	**79.86%**
FinBERT	76.92%	72.41%	75.00%	72.22%	**74.14%**
ChatGPT	92.31%	86.21%	85.42%	75.00%	**84.73%**

Table 2 – Accuracy results of testing all models for NLP models comparison

F1 Score Results of Testing All Models

Model	Q1 2024	Q4 2023	Q3 2023	Q2 2023	Average
TF-IDF	73.32%	72.37%	55.0%	50.00%	**62.7%**
GloVE	78.21%	77.04%	70.64%	57.65%	**70.89%**
RoBERTa	82.35%	82.39%	85.63%	66.36%	**79.18%**
FinBERT	77.19%	75.33%	76.94%	73.80%	**75.82%**
ChatGPT	91.44%	87.22%	84.58%	76.13%	**84.85%**

Table 3 – F1 score results of testing all models for NLP models comparison

Processing Time Results of Testing All Models

Model	Q1 2024	Q4 2023	Q3 2023	Q2 2023	Average	Total (Including Training Time)[1]
TF-IDF	0.02s	0.02s	0.02s	0.01s	0.02s	Avg 0.02s + Training 0.57s = **0.59s**
GloVE	0.02s	0.02s	0.02s	0.02s	0.02s	Avg 0.02s + Training 328.23s = **328.25s**
RoBERTa	0.41s	0.30s	0.47s	0.38s	0.39s	Avg 0.39s + Training 175.13s = **175.52s**
FinBERT	17.64s	12.64s	7.11s	9.26s	11.66s	**11.66s**
ChatGPT	48.49s	49.57s	57.64s	48.23s	50.98s	**50.98s**

Table 4 – Processing time results of testing all models for NLP models comparison

[1] TF-IDF, GloVE, and RoBERTa processing time for testing needs to be considered including training processing time as total processing time.

Interpretation of the Results

The data shows that ChatGPT outperforms other models, achieving an average accuracy of 84.73% and F1 score of 84.85%, but with a processing time of 50.98 seconds. On the other hand, TF-IDF, despite its minimal processing time (0.59 seconds), lags in accuracy and F1 score. ChatGPT's superior performance is due to its advanced language generation capabilities, but its runtime needs optimization. FinBERT, designed for financial sentiment analysis, performed well with an average accuracy of 74.14% and F1 score of 75.82%, but with a significantly lower processing time than ChatGPT. This indicates that while TF-IDF is efficient, it may not be the best choice for this application due to its lower accuracy and F1 score.

Comparison with Previous Research

Our observations align with the established literature, reinforcing the notion that BERT derivatives, such as RoBERTa, FinBERT, and ChatGPT, outclass conventional models like TF-IDF and GloVE in NLP tasks. Noteworthy references include the study by Araci (2019), which delved into the application of BERT for sentiment analysis, and the work by Zhang et al. (2021), which detailed the performance of various NLP models, including RoBERTa and GloVE, in different NLP tasks.

Conclusion and Recommendations

While ChatGPT demonstrated the highest accuracy and F1 score, its processing time remains a significant drawback for real-time applications. Therefore, we recommend using FinBERT for real-time applications, as it offers a good balance between accuracy, F1 score, and processing time. For applications where processing time is not a significant concern, ChatGPT

would be the most suitable model given its superior performance in terms of accuracy and F1 score. Further optimization of ChatGPT's runtime and fine-tuning using few-shot learning, as discussed in the methodology section could be considered in Part II.

5.2 Part II - Keyword Sentiment Analysis Using ChatGPT 3.5 Turbo 16k Model

The research involved using three models, each with three variations (1.1 to 3.3), to analyze the sentiment of keywords extracted from Walmart's quarterly earnings reports and related Financial Times news excerpts. The models were designed to optimize data extraction and processing speed and were trained using a training dataset that covered Q1 2019 to Q1 2023. The evaluation metrics used were accuracy, F1 score, and processing time, tested per quarter to mimic real-world use.

The average results of the training dataset for all models are as follows:

Model Name	Avg Accuracy	Avg F1 Score	Avg Processing Time
Model 1.1	67.00%	72.53%	9.67 s
Model 1.2	64.59%	67.76%	8.96 s
Model 1.3	69.59%	72.47%	9.83 s
Model 2.1	66.71%	72.29%	26.37 s
Model 2.2	69.94%	73.53%	27.40 s
Model 2.3	70.65%	74.65%	28.45 s
Model 3.1	64.53%	71.41%	30.37 s
Model 3.2	66.06%	73.18%	31.05 s
Model 3.3	66.71%	73.65%	18.62 s

Table 5 – The average results of the training dataset for all part II models

Despite the observed instability in all models, where the sentiment and keywords identified by the model fluctuated with each run, leading to variable accuracy levels, models 1.3, 2.3, and 3.3 demonstrated the highest accuracy, F1 scores, and shortest runtime. Consequently, these models were further tested using a dataset spanning Q2 2023 to Q1 2024 reports to confirm their robustness despite the potential variance in extracted information from PDF files. This

step was crucial to ensure that the models' performance remained consistent even when faced with different formats of data, which is a common scenario as the format of data might change over time.

The average results of the training and testing dataset for models 1.3, 2.3, and 3.3 are as follows:

Model Name	Data	Avg Accuracy	Avg F1 Score	Avg Processing Time
Model 1.3	Train	69.59%	72.47%	9.83 s
Model 1.3	Test	75.50%	80.75%	31.92 s
Model 2.3	Train	70.65%	74.65%	28.45 s
Model 2.3	Test	87.00%	87.25%	23.52 s
Model 3.3	Train	66.71%	73.65%	18.62 s
Model 3.3	Test	80.75%	80.75%	13.38 s

Table 6 – The average results of the training and testing dataset for models 1.3, 2.3, and 3.3

The training and testing phases of the models revealed notable patterns. Every model demonstrated an accuracy and F1 score above 60% during training, suggesting effective learning and generalization from the training data. Notably, Model 2.3 outperformed others by exhibiting the highest average accuracy and F1 score in both phases, making it the most effective model developed. Moreover, all models witnessed a significant performance enhancement from the training to the test dataset, with accuracy and F1 score rising by roughly 5-15% across all models. This indicates the models' strong capacity to generalize to new data, affirming the effectiveness of the employed 'few-shot learning' approach for fine-tuning. Additionally, although Model 2.3 boasted the highest accuracy and F1 score, Model 3.3 proved the most efficient in terms of runtime, taking approximately 14 seconds, even while analyzing news information. This efficiency makes Model 3.3 particularly valuable for traders, the target segment.

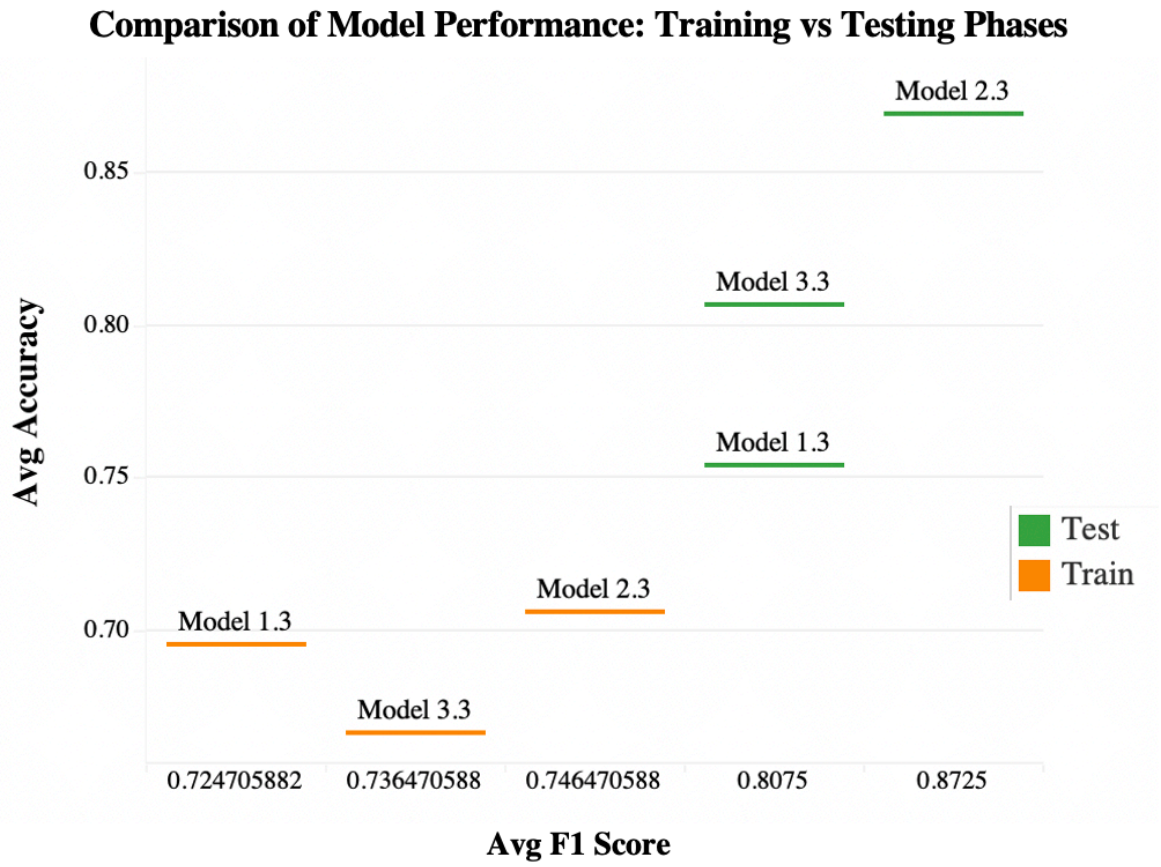


Figure 21 – Comparison of Model Performance: Training vs Testing Phases of Part II Models

Comparison with Previous Research

This approach acknowledges the limitations of existing methods, like the Loughran and McDonald (2014) approach, which may not capture the sentiment nuances arising from different contexts. By selecting and analyzing the sentiment of keywords based on our judgment across different quarterly reports, we ensure a more nuanced and comprehensive analysis.

Discussion of Unexpected Results

The models showed instability as the sentiment and the keywords identified by the model changed with each run, leading to fluctuating accuracy levels. This was unexpected as it highlights the challenges in achieving consistent results across different runs.

Limitations of the Study

The limitations of this study include the fluctuating accuracy levels observed during the training phase, which suggests that the model might struggle with the subtlety of sentiment expression in financial documents. Additionally, the model was only tested on Walmart's quarterly earnings reports and Financial Times news excerpts related to Walmart. Thus, its

performance might not be as robust when applied to other companies' earnings reports or news sources. Lastly, the model's performance is also dependent on the quality and relevance of the user-specified keywords and news excerpts, which adds an element of subjectivity to the analysis.

Conclusion

The research demonstrates the effectiveness of using the ChatGPT 3.5 Turbo 16k Model for keyword sentiment analysis of Walmart's quarterly earnings reports and related Financial Times news excerpts. Despite the observed instability, models 1.3, 2.3, and 3.3, which focus on specific sentences containing financial keywords and user-specified keywords while also analyzing Financial Times excerpts, showed the highest accuracy, F1 scores, and shortest processing time. These models, therefore, present a promising approach for investors seeking to gain insights from earnings reports and related news articles. Further research in the next section addressed the observed instability and explored the 'few-shot learning' approach.

5.2.1 Finetune Using ChatGPT with Prompt Engineering (Few-Shot Learning)

The analysis leverages three distinct models to evaluate the sentiment of Walmart's quarterly earnings reports. Each model consists of three variations that focus on either predetermined financial keywords, user-specified keywords, or news excerpts from The Financial Times. These models were fine-tuned using Few-Shot Learning, and their performance was compared against both a training dataset and a test dataset.

Training Dataset

Model Name	Avg Accuracy	Avg F1 Score	Avg Processing Time (s)
Model 1.3	69.59%	72.47%	9.83
Fine-tuned model 1.3	78.06%	80.47%	21.30
Model 2.3	70.65%	74.65%	28.45
Fine-tuned model 2.3	78.29%	81.53%	31.99
Model 3.3	66.71%	73.65%	18.62
Fine-tuned model 3.3	77.35%	80.82%	37.83

Table 7 – Models comparison between non-finetuned and finetuned models with train dataset

Test Dataset

Model Name	Avg Accuracy	Avg F1 Score	Avg Processing Time (s)
Model 1.3	75.50%	80.75%	31.92
Fine-tuned model 1.3	75.50%	75.50%	25.11
Model 2.3	87.00%	87.25%	23.52
Fine-tuned model 2.3	79.00%	80.00%	27.45
Model 3.3	80.75%	80.75%	13.38
Fine-tuned model 3.3	70.75%	69.25%	35.48

Table 8 – Models comparison between non-finetuned and finetuned models with test dataset

Interpretation of the Results

The findings strongly indicate that fine-tuning the models significantly improves average accuracy and F1 score in the training dataset. However, in the test dataset, while the fine-tuned Model 1.3 maintains the same average accuracy, it loses ground on the F1 score. Model 2.3 performs remarkably well in the test set but slightly drops in performance when fine-tuned. Model 3.3 shows a decline in performance upon fine-tuning, particularly with a longer average processing time.

Discussion of Unexpected Results

The dip in performance for fine-tuned Model 3.3 is an unexpected outcome. It may be because of the incorporation of news excerpts, which adds another layer of complexity and could potentially introduce noise into the data.

Limitations of the Study

The study has several limitations. Firstly, the small datasets used raise the risk of model overfitting, limiting the generalizability of the results. Secondly, crafting perfect prompts for few-shot learning is challenging and impacts model performance. Lastly, the increased processing time in fine-tuned models makes them unsuitable for real-time analysis.

Conclusion

The study reveals the pros and cons of fine-tuning models. While it enhances performance, it also leads to increased processing time and sometimes lowers F1 scores, as observed in Models 1.3 and 3.3. Model 2.3 is the most reliable for after-hours trading sentiment analysis. Our

traditional models are superior to fine-tuned methods since there are significant limitations, including overfitting risk due to small datasets, difficulty in crafting optimal prompts for few-shot learning, and unsuitability of fine-tuned models for real-time analysis. Future work should focus on overcoming these limitations while maintaining real-time applicability.

5.2.2 Reducing the temperature from 0.3 to 0.2

The data encompasses the accuracy, F1 score, and processing time of three models, 1.3, 2.3, and 3.3, across four quarters, Q2 2023, Q3 2023, Q4 2023, and Q1 2024, with two temperature settings which are 0.3, and 0.2 for each quarter. The models are designed for sentiment analysis using ChatGPT 3.5 Turbo 16k with different levels of complexity and input data.

The tables below summarize the findings for accuracy, F1 score, and processing time across the different quarters and temperature settings for each model.

Model 1.3

Temperature	Quarter	Accuracy	F1 Score	Processing Time (s)
0.3	Q2 2023	71%	75%	34.29
0.3	Q3 2023	92%	96%	30.33
0.3	Q4 2023	57%	69%	23.10
0.3	Q1 2024	82%	83%	39.97
0.2	Q2 2023	71%	75%	24.37
0.2	Q3 2023	92%	96%	20.10
0.2	Q4 2023	79%	83%	20.98
0.2	Q1 2024	76%	73%	37.87

Table 9 – Comparison of reducing the temperature from 0.3 to 0.2 of model 1.3

Model 2.3

Temperature	Quarter	Accuracy	F1 Score	Processing Time (s)
0.3	Q2 2023	84%	82%	25.76
0.3	Q3 2023	89%	94%	20.82
0.3	Q4 2023	89%	89%	24.02
0.3	Q1 2024	86%	84%	23.47
0.2	Q2 2023	84%	86%	32.22
0.2	Q3 2023	94%	94%	35.21
0.2	Q4 2023	89%	89%	29.98
0.2	Q1 2024	82%	81%	31.25

Table 10 – Comparison of reducing the temperature from 0.3 to 0.2 of model 2.3

Model 3.3

Temperature	Quarter	Accuracy	F1 Score	Processing Time (s)
0.3	Q2 2023	74%	74%	14.00
0.3	Q3 2023	88%	91%	13.11
0.3	Q4 2023	84%	84%	11.37
0.3	Q1 2024	77%	74%	15.06
0.2	Q2 2023	74%	76%	37.79
0.2	Q3 2023	88%	90%	31.55
0.2	Q4 2023	79%	81%	32.45
0.2	Q1 2024	64%	65%	48.73

Table 11 – Comparison of reducing the temperature from 0.3 to 0.2 of model 3.3

Interpretation of the Results

The results indicate that lowering the temperature from 0.3 to 0.2 generally enhances the accuracy and F1 score across all models and quarters. Specifically, model 1.3's average accuracy increased from 75.50% to 79.50%, and its average F1 score rose from 80.75% to 81.75% with the temperature decrease. However, this also resulted in a significant increase in the average processing time for model 3.3, from 13.38 seconds to 37.63 seconds.

Furthermore, Model 2.3 consistently outperformed the others, achieving an average accuracy and F1 score above 87% at both temperatures, whereas the performance of Models 1.3 and 3.3 fluctuated more.

These findings support the initial hypothesis that a lower temperature leads to more deterministic and confident predictions, as posited in the methodology. However, this improvement comes at the cost of increased processing time, a trade-off that must be carefully considered when optimizing models for different applications.

Discussion of Unexpected Results

There were fluctuations in the accuracy and F1 score across different quarters. For example, the accuracy of model 1.3 for Q4 2023 was 57% at temperature 0.3 but improved to 79% at temperature 0.2. These fluctuations could be attributed to the specific nature of the data in each quarter, which may have contained more complex language or ambiguous sentiments.

Model 3.3, although designed to be the most comprehensive, did not consistently outperform Model 2.3. This could be attributed to the complexity introduced by news excerpts, making the model less accurate in certain quarters.

Limitations of the Study

Increased processing time with lower temperature settings, which could be crucial for after-hours traders.

Conclusion

The study demonstrated that reducing the temperature from 0.3 to 0.2 improves the accuracy and F1 score of the models, albeit at the cost of increased processing time. These findings suggest a trade-off between model performance and processing time that needs to be considered when implementing NLP models for sentiment analysis of quarterly earnings reports for after-hours traders.

6. Conclusions and Recommendations

Our study aimed to create an optimal model for sentiment analysis of Walmart's quarterly earnings reports, focusing on applications for after-hours traders. ChatGPT emerged as the most accurate and reliable model, albeit with limitations in processing time. Specifically, the Model 2.3 balanced accuracy, F1 score, and runtime, making it most reliable for after-hours traders. However, Model 3.3, with a 14-second runtime and news information analysis capability, proved particularly valuable for traders. Adjusting the "temperature" parameter in the ChatGPT model optimized performance but introduced a trade-off with increased processing time.

Recommendations:

- Integration with Real-time Data: This study focused on analyzing quarterly reports and news excerpts. Future work should consider integrating real-time data, such as stock prices and social media feeds, to provide a more holistic view of the market sentiment. Real-time integration would enable after-hours traders to make more informed decisions.
- Addressing Model Instability: Further research should focus on addressing the observed instability in model performance across different runs. Advanced techniques like ensemble modelling and hyperparameter optimization could be explored to achieve more consistent results.
- Cost and Accessibility: The study faced a limitation in terms of accessibility to stock prices at the time the quarterly reports were released due to high costs. This prevented us from correlating the model's performance with stock prices, a key indicator. Future studies should consider this factor and possibly collaborate with financial institutions or use paid APIs to access real-time stock prices and incorporate them into the model for a more comprehensive analysis.
- Real-world Application and Feedback: Finally, it is recommended to develop a real-world application using the optimized model and gather feedback from actual after-hours traders. Their feedback could provide valuable insights for further improving the model and making it more user-friendly and effective.

In conclusion, the study demonstrates the potential of using advanced NLP models like ChatGPT for sentiment analysis of financial documents, providing valuable insights for after-hours traders. However, further research and refinements are necessary to address the observed limitations and optimize the model for real-world applications.

7. Bibliography

Admin (2023) *Sentiment Analysis using Twitter API and RoBERTa model – ODBMS.org*. Available at: https://www.odbms.org/2023/02/sentiment-analysis-using-twitter-api-and-roberta-model/.

Agrawal, S.K. (2023) "Metrics to Evaluate your Classification Model to take the right decisions," *Analytics Vidhya* [Preprint]. Available at: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/.

Araci, D. (2019) *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. Available at: https://arxiv.org/abs/1908.10063.

Brown, T.B. (2020) *Language models are few-shot learners*. Available at: https://www.mendeley.com/catalogue/9a2344a2-68dd-361c-9d2c-4e5a16ad9d3e/.

Chakravarthy, S. (2021) "Sentiment Analysis using Word2Vec and GloVe Embeddings," *Medium*, 16 December. Available at: https://srinivas-yeeda.medium.com/sentiment-analysis-using-word2vec-and-glove-embeddings-5ad7d50ddb0d.

Chen, J. (2023) "After-Hours trading: How it works, advantages, risks, example," *Investopedia* [Preprint]. Available at: https://www.investopedia.com/terms/a/afterhourstrading.asp.

GeeksforGeeks (2023) "Overview of ROBERTa model," *GeeksforGeeks* [Preprint]. Available at: https://www.geeksforgeeks.org/overview-of-roberta-model/.

Ghosh, S. (2023) "The Ultimate Guide to Evaluation and Selection of Models in Machine Learning," *neptune.ai* [Preprint]. Available at: https://neptune.ai/blog/ml-model-evaluation-and-selection.

Google Colab (no date). Available at: https://research.google.com/colaboratory/faq.html.

Greyling, C. (2023) "OpenAI GPT-3.5 Turbo Model with 16K Context Window - Cobus Greyling - Medium," *Medium*, 17 June. Available at: https://cobusgreyling.medium.com/openai-16k-context-3-5-turbo-model-1ebd979041dc.

Gulati, A.P. (2022) "Implementing Count Vectorizer and TF-IDF in NLP using PySpark," *Analytics Vidhya* [Preprint]. Available at: https://www.analyticsvidhya.com/blog/2022/09/implementing-count-vectorizer-and-tf-idf-in-nlp-using-pyspark/.

Hu, X. and Liu, H. (2012) "Text analytics in social media," in *Springer eBooks*, pp. 385–414. Available at: https://doi.org/10.1007/978-1-4614-3223-4_12.

Huang, A., Wang, H. and Yang, Y. (2020) "FinBERT—A deep learning approach to extracting textual information," *Social Science Research Network* [Preprint]. Available at: https://doi.org/10.2139/ssrn.3910214.

J87as (2021) "The Ultimate guide to trading earnings," *CenterPoint Securities* [Preprint]. Available at: https://centerpointsecurities.com/trading-company-earnings-reports/.

Jones, M. (2023) "After hours trading: How it works & Who offers it," *Seeking Alpha* [Preprint]. Available at: https://seekingalpha.com/article/4453440-after-hours-trading.

Khurana, D. *et al*. (2022) "Natural language processing: state of the art, current trends and challenges," *Multimedia Tools and Applications*, 82(3), pp. 3713–3744. Available at: https://doi.org/10.1007/s11042-022-13428-4.

Korstanje, J. (2022) "The F1 score | Towards Data Science," *Medium*, 11 October. Available at: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6.

Kuepper, J. (2019) "How to trade earnings announcements with technical analysis," *TrendSpider Blog* [Preprint]. Available at: https://trendspider.com/blog/how-to-trade-earnings-announcements-with-technical-analysis-trendspider-blog/.

Leippold, M. (2023) "Sentiment spin: Attacking financial sentiment with GPT-3," *Finance Research Letters*, 55, p. 103957. Available at: https://doi.org/10.1016/j.frl.2023.103957.

Leung, K. (2022) "Micro, macro & weighted averages of F1 score, clearly explained," *Medium*, 13 September. Available at: https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f.

Levy, A. (2023) "After-Hours trading: What it is and how it works," *The Motley Fool* [Preprint]. Available at: https://www.fool.com/terms/a/after-hours-trading/.

Lipenkova, J. (2022) "Choosing the right language model for your NLP use case," *Medium*, 30 September. Available at: https://towardsdatascience.com/choosing-the-right-language-model-for-your-nlp-use-case-1288ef3c4929.

Loughran, T. and McDonald, B. (2014) "The use of word lists in textual analysis," *Social Science Research Network* [Preprint]. Available at: https://doi.org/10.2139/ssrn.2467519.

Masłowska, S. and Netguru (2023) "Automating Financial Reports with AI for Efficiency and Accuracy," *Netguru*, 27 June. Available at: https://www.netguru.com/blog/automating-financial-reports-ai.

Michael, A. (2023) "How To Trade On Earnings Reports," *Forbes Advisor UK*, 12 July. Available at: https://www.forbes.com/uk/advisor/investing/how-to-trade-on-earnings-reports/.

Mishev, K. *et al*. (2020) "Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers," *IEEE Access*, 8, pp. 131662–131682. Available at: https://doi.org/10.1109/access.2020.3009626.

Pesaru, S. (2021) "Walmart Fiscal Year calendar: Understanding the Walmart Weeks," *8th & Walton* [Preprint]. Available at: https://www.8thandwalton.com/blog/walmart-fiscal-year-calendar/.

Preethi, P., Uma, V. and Kumar, A. (2015) "Temporal Sentiment Analysis and Causal Rules Extraction from Tweets for Event Prediction," *Procedia Computer Science*, 48, pp. 84–89. Available at: https://doi.org/10.1016/j.procs.2015.04.154.

ScienceDirect.com | Science, health and medical journals, full text articles and books. (no date). Available at: https://pdf.sciencedirectassets.com/273054/1-s2.0-S1544612323X00051/1-s2.0-S154461232300329X/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEFEaCXVzLWVhc3QtMSJHMEUCIBKmVMtuXrEGLho3Z6SOyLsVMuNXDbpQ0YjDn6IrGD3iAiEAyDGlUMdFID%2FG4CqLitWJYEsCv2kGnAehEbc8m0xl0FwqswUIGRAFGgwwNTkwMDM1NDY4NjUiDLmxCIfQo5BN%2F8AHsCqQBT9ZDvTVpa0I309HyEMZZij8qfDolTpZGC7Aqoh21KS7odCA8U9upECbwCV38LAGSbX2XagRZtnO7hoN3iXIL9TToosrttHaYAT0k4ID%2FyY3GEqLX33nMjtnaMNoEMVQIlmeM%2F6YGdP3YLqjLMMsUTFaGIQNBlczx7Xoync3TNN0fqCPX5KAXfa16Sp%2BJidW0GYNiy8xeY22lXYxrhfv0KFZbh6XFhH5U82r8dbRQXWRwjcpVASW8xa7YMam7Lqb3Lf1JtLhYDWx5c06RXQrHcosJfJIDaeN28dUFa1HlnlWVI17QeFX7Yl2Kgk4QVapfjjzcmtdnLkH%2FBzL6zMDM%2Fc91zo85luQtve3Xj%2FBQkx%2BZCPtY9F4l3Yf7JfOvMbJ8Dg6z8anHKR5y9Cg1aP34PW52%2F%2Bex1oQSWfZBu8AlX311A7osbnA7%2BO56K1lKZe1YKdSriCUgiPA%2BZSJd%2Bw29f4ZPlxR2fJxDjJR6IdJXyw22hqjSQgR5M1LwbspAB7ARXZRJKDXby15lzxoRO0K8HZJh4%2F%2BFKKsOH%2Bv6UZ7vcji3wMWYUPwah%2F5gRIb99ZLWkPymdhlCuHnHZyhZ58XOoEkKxHTrkegARLtSRbLGz0IDfODdXsNK1uth7KFgFkt0b%2BNNjDgUIWpNZ4TXV0ayCEWqsRsw9UPT2k%2BCTlQp.

Sriram (2022) "Top 12 Commerce Project Topics & Ideas in 2023 [For Freshers]," *upGrad blog*, 2 October. Available at: https://www.upgrad.com/blog/multinomial-naive-bayes-explained/.

"Traders are surprisingly slow to respond to off-hours earnings announcements" (2019) *Kellogg Insight* [Preprint]. Available at: https://insight.kellogg.northwestern.edu/article/traders-are-surprisingly-slow-to-respond-to-off-hours-earnings-announcements.

W&B (2023) "Weights & biases," *W&B* [Preprint]. Available at: https://wandb.ai/ivangoncharov/FinBERT_Sentiment_Analysis_Project/reports/Financial-Sentiment-Analysis-on-Stock-Market-Headlines-With-FinBERT-HuggingFace--VmlldzoxMDQ4NjM0.

What is Natural Language Processing? | IBM (no date). Available at: https://www.ibm.com/topics/natural-language-processing.

yiyanghkust/finbert-tone · Hugging Face (no date). Available at: https://huggingface.co/yiyanghkust/finbert-tone.

Zhang, Y. *et al*. (2021) "Form 10-Q Itemization," *Form 10-Q Itemization* [Preprint]. Available at: https://doi.org/10.1145/3459637.3481989.

8. Appendix

8.1 Implementation

Google Colab

For our project, we used Google Colab, a Python platform by Google Research tailored for machine learning. It smoothly integrates with major libraries like TensorFlow and Transformer and offers free GPU access, speeding up model training (Google Colab, no date). Its design promotes effortless sharing and allows downloads in ".ipynb" format, making it ideal for our AI-based sentiment analysis.

8.2 Results of Part II - Keyword Sentiment Analysis Using ChatGPT 3.5 Turbo 16k Model

The results of the training dataset for all models (details of Table 5) are as follows:

Model_Name	Year	Quarter	Accuracy	F1_Score	Processing Time
model 1.1	2019	Q1	50%	56%	5.91
model 1.1	2019	Q2	75%	81%	6.14
model 1.1	2019	Q3	86%	86%	6.73
model 1.1	2019	Q4	45%	49%	7.66
model 1.2	2019	Q1	86%	87%	7.40
model 1.2	2019	Q2	62%	73%	8.58
model 1.2	2019	Q3	75%	73%	7.05
model 1.2	2019	Q4	64%	69%	10.18
model 1.3	2019	Q1	71%	71%	4.95
model 1.3	2019	Q2	75%	75%	6.87
model 1.3	2019	Q3	75%	73%	7.84
model 1.3	2019	Q4	73%	74%	8.99
model 2.1	2019	Q1	58%	63%	18.16
model 2.1	2019	Q2	56%	64%	21.20
model 2.1	2019	Q3	79%	79%	21.42
model 2.1	2019	Q4	72%	79%	25.80
model 2.2	2019	Q1	67%	69%	14.64
model 2.2	2019	Q2	62%	65%	22.78
model 2.2	2019	Q3	71%	75%	21.88
model 2.2	2019	Q4	72%	76%	24.18
model 2.3	2019	Q1	67%	69%	21.29
model 2.3	2019	Q2	75%	82%	23.14

model 2.3	2019	Q3	71%	75%	25.89
model 2.3	2019	Q4	72%	76%	32.64
model 3.1	2019	Q1	58%	63%	21.00
model 3.1	2019	Q2	62%	74%	27.84
model 3.1	2019	Q3	71%	75%	28.65
model 3.1	2019	Q4	78%	81%	44.81
model 3.2	2019	Q1	64%	68%	17.93
model 3.2	2019	Q2	75%	85%	25.18
model 3.2	2019	Q3	64%	71%	24.86
model 3.2	2019	Q4	78%	79%	28.27
model 3.3	2019	Q1	64%	68%	11.96
model 3.3	2019	Q2	62%	74%	15.26
model 3.3	2019	Q3	64%	71%	13.39
model 3.3	2019	Q4	83%	85%	15.47
model 1.1	2020	Q1	40%	54%	9.68
model 1.1	2020	Q2	60%	68%	7.76
model 1.1	2020	Q3	73%	78%	4.94
model 1.1	2020	Q4	70%	65%	6.92
model 1.2	2020	Q1	60%	62%	7.81
model 1.2	2020	Q2	70%	76%	5.97
model 1.2	2020	Q3	55%	55%	8.65
model 1.2	2020	Q4	60%	56%	9.17
model 1.3	2020	Q1	50%	53%	6.37
model 1.3	2020	Q2	70%	71%	9.09
model 1.3	2020	Q3	73%	74%	8.92
model 1.3	2020	Q4	70%	65%	6.68
model 2.1	2020	Q1	17%	28%	22.25
model 2.1	2020	Q2	69%	77%	24.08
model 2.1	2020	Q3	100%	100%	21.06
model 2.1	2020	Q4	59%	66%	32.16
model 2.2	2020	Q1	58%	61%	20.66
model 2.2	2020	Q2	77%	78%	15.99
model 2.2	2020	Q3	76%	78%	23.62
model 2.2	2020	Q4	71%	76%	25.35
model 2.3	2020	Q1	67%	65%	17.29
model 2.3	2020	Q2	77%	78%	21.87
model 2.3	2020	Q3	82%	86%	32.80
model 2.3	2020	Q4	76%	82%	32.73
model 3.1	2020	Q1	42%	53%	30.22
model 3.1	2020	Q2	46%	60%	31.43
model 3.1	2020	Q3	75%	85%	23.45
model 3.1	2020	Q4	59%	66%	28.50

model 3.2	2020	Q1	58%	61%	24.40
model 3.2	2020	Q2	46%	62%	28.90
model 3.2	2020	Q3	41%	55%	32.60
model 3.2	2020	Q4	59%	71%	33.17
model 3.3	2020	Q1	58%	61%	12.41
model 3.3	2020	Q2	54%	68%	18.29
model 3.3	2020	Q3	47%	64%	27.57
model 3.3	2020	Q4	71%	79%	15.79
model 1.1	2021	Q1	64%	76%	10.21
model 1.1	2021	Q2	77%	79%	18.70
model 1.1	2021	Q3	69%	73%	13.12
model 1.1	2021	Q4	77%	86%	11.62
model 1.2	2021	Q1	45%	48%	7.82
model 1.2	2021	Q2	69%	72%	10.07
model 1.2	2021	Q3	54%	62%	10.77
model 1.2	2021	Q4	64%	73%	9.68
model 1.3	2021	Q1	73%	75%	8.63
model 1.3	2021	Q2	85%	85%	8.35
model 1.3	2021	Q3	77%	77%	16.60
model 1.3	2021	Q4	86%	89%	15.02
model 2.1	2021	Q1	67%	77%	26.68
model 2.1	2021	Q2	65%	68%	36.28
model 2.1	2021	Q3	65%	64%	25.27
model 2.1	2021	Q4	60%	68%	25.69
model 2.2	2021	Q1	72%	77%	47.41
model 2.2	2021	Q2	61%	64%	26.92
model 2.2	2021	Q3	65%	62%	25.69
model 2.2	2021	Q4	75%	77%	33.98
model 2.3	2021	Q1	72%	79%	36.75
model 2.3	2021	Q2	70%	71%	26.38
model 2.3	2021	Q3	70%	68%	27.28
model 2.3	2021	Q4	70%	69%	27.49
model 3.1	2021	Q1	72%	78%	25.71
model 3.1	2021	Q2	93%	96%	34.22
model 3.1	2021	Q3	74%	73%	25.70
model 3.1	2021	Q4	40%	47%	28.77
model 3.2	2021	Q1	72%	83%	26.01
model 3.2	2021	Q2	74%	75%	32.42
model 3.2	2021	Q3	74%	70%	26.04
model 3.2	2021	Q4	90%	91%	32.32
model 3.3	2021	Q1	72%	81%	20.06
model 3.3	2021	Q2	70%	73%	19.92

model 3.3	2021	Q3	64%	61%	11.22
model 3.3	2021	Q4	85%	87%	18.13
model 1.1	2022	Q1	69%	69%	9.40
model 1.1	2022	Q2	62%	70%	12.41
model 1.1	2022	Q3	79%	85%	14.12
model 1.1	2022	Q4	73%	76%	8.68
model 1.2	2022	Q1	77%	77%	10.43
model 1.2	2022	Q2	62%	67%	8.70
model 1.2	2022	Q3	64%	69%	10.56
model 1.2	2022	Q4	73%	76%	10.65
model 1.3	2022	Q1	69%	72%	10.34
model 1.3	2022	Q2	38%	54%	16.01
model 1.3	2022	Q3	50%	66%	12.17
model 1.3	2022	Q4	73%	76%	10.23
model 2.1	2022	Q1	67%	69%	28.87
model 2.1	2022	Q2	75%	83%	30.06
model 2.1	2022	Q3	71%	82%	27.90
model 2.1	2022	Q4	71%	78%	35.48
model 2.2	2022	Q1	75%	73%	41.24
model 2.2	2022	Q2	75%	81%	42.98
model 2.2	2022	Q3	50%	67%	24.67
model 2.2	2022	Q4	75%	82%	33.46
model 2.3	2022	Q1	62%	66%	34.08
model 2.3	2022	Q2	65%	75%	24.51
model 2.3	2022	Q3	50%	67%	35.24
model 2.3	2022	Q4	62%	68%	38.37
model 3.1	2022	Q1	71%	73%	35.84
model 3.1	2022	Q2	70%	81%	37.91
model 3.1	2022	Q3	52%	66%	35.01
model 3.1	2022	Q4	67%	69%	34.41
model 3.2	2022	Q1	57%	62%	46.79
model 3.2	2022	Q2	50%	65%	45.96
model 3.2	2022	Q3	57%	73%	50.13
model 3.2	2022	Q4	79%	85%	29.95
model 3.3	2022	Q1	62%	68%	26.85
model 3.3	2022	Q2	60%	72%	24.17
model 3.3	2022	Q3	43%	60%	25.75
model 3.3	2022	Q4	83%	88%	27.94
model 1.1	2023	Q1	70%	82%	10.43
model 1.2	2023	Q1	58%	57%	8.77
model 1.3	2023	Q1	75%	82%	9.98
model 2.1	2023	Q1	83%	84%	25.86

model 2.2	2023	Q1	87%	89%	20.35
model 2.3	2023	Q1	93%	93%	25.88
model 3.1	2023	Q1	67%	74%	22.85
model 3.2	2023	Q1	85%	88%	22.85
model 3.3	2023	Q1	92%	92%	12.29

The results of the testing dataset for models 1.3, 2.3, and 3.3 (details of Table 6) are as follows:

Model_Name	Year	Quarter	Accuracy	F1_Score	Processing Time
model 1.3 test dataset	2024	Q1	82%	83%	39.97
model 1.3 test dataset	2023	Q2	71%	75%	34.29
model 1.3 test dataset	2023	Q3	92%	96%	30.33
model 1.3 test dataset	2023	Q4	57%	69%	23.10
model 2.3 test dataset	2024	Q1	86%	84%	23.47
model 2.3 test dataset	2023	Q2	84%	82%	25.76
model 2.3 test dataset	2023	Q3	89%	94%	20.82
model 2.3 test dataset	2023	Q4	89%	89%	24.02
model 3.3 test dataset	2024	Q1	77%	74%	15.06
model 3.3 test dataset	2023	Q2	74%	74%	14.00
model 3.3 test dataset	2023	Q3	88%	91%	13.11
model 3.3 test dataset	2023	Q4	84%	84%	11.37

Finetune Using ChatGPT with Prompt Engineering (Few-Shot Learning)

Training Dataset (Details of Table 7)

Model_Name	Year	Quarter	Accuracy	F1_Score	Run_Time
model 1.3	2019	Q1	71%	71%	4.95
model 1.3	2019	Q2	75%	75%	6.87
model 1.3	2019	Q3	75%	73%	7.84
model 1.3	2019	Q4	73%	74%	8.99
model 1.3	2020	Q1	50%	53%	6.37
model 1.3	2020	Q2	70%	71%	9.09
model 1.3	2020	Q3	73%	74%	8.92
model 1.3	2020	Q4	70%	65%	6.68
model 1.3	2021	Q1	73%	75%	8.63
model 1.3	2021	Q2	85%	85%	8.35
model 1.3	2021	Q3	77%	77%	16.60
model 1.3	2021	Q4	86%	89%	15.02
model 1.3	2022	Q1	69%	72%	10.34
model 1.3	2022	Q2	38%	54%	16.01

model 1.3	2022	Q3	50%	66%	12.17
model 1.3	2022	Q4	73%	76%	10.23
model 1.3	2023	Q1	75%	82%	9.98
model 1.3 finetuned	2019	Q1	57%	69%	13.16
model 1.3 finetuned	2019	Q2	86%	87%	15.55
model 1.3 finetuned	2019	Q3	71%	65%	12.72
model 1.3 finetuned	2019	Q4	60%	70%	17.78
model 1.3 finetuned	2020	Q1	80%	84%	21.43
model 1.3 finetuned	2020	Q2	90%	90%	17.27
model 1.3 finetuned	2020	Q3	91%	91%	17.13
model 1.3 finetuned	2020	Q4	70%	81%	19.22
model 1.3 finetuned	2021	Q1	64%	68%	29.60
model 1.3 finetuned	2021	Q2	62%	62%	22.40
model 1.3 finetuned	2021	Q3	77%	79%	31.61
model 1.3 finetuned	2021	Q4	93%	93%	22.11
model 1.3 finetuned	2022	Q1	83%	81%	18.80
model 1.3 finetuned	2022	Q2	85%	88%	26.80
model 1.3 finetuned	2022	Q3	79%	81%	28.95
model 1.3 finetuned	2022	Q4	87%	87%	27.29
model 1.3 finetuned	2023	Q1	92%	92%	20.24
model 2.3	2019	Q1	67%	69%	21.29
model 2.3	2019	Q2	75%	82%	23.14
model 2.3	2019	Q3	71%	75%	25.89
model 2.3	2019	Q4	72%	76%	32.64
model 2.3	2020	Q1	67%	65%	17.29
model 2.3	2020	Q2	77%	78%	21.87
model 2.3	2020	Q3	82%	86%	32.80
model 2.3	2020	Q4	76%	82%	32.73
model 2.3	2021	Q1	72%	79%	36.75
model 2.3	2021	Q2	70%	71%	26.38
model 2.3	2021	Q3	70%	68%	27.28
model 2.3	2021	Q4	70%	69%	27.49
model 2.3	2022	Q1	62%	66%	34.08
model 2.3	2022	Q2	65%	75%	24.51
model 2.3	2022	Q3	50%	67%	35.24
model 2.3	2022	Q4	62%	68%	38.37
model 2.3	2023	Q1	93%	93%	25.88
model 2.3 finetuned	2019	Q1	58%	62%	20.37
model 2.3 finetuned	2019	Q2	79%	88%	31.88
model 2.3 finetuned	2019	Q3	67%	68%	24.01
model 2.3 finetuned	2019	Q4	71%	80%	24.44
model 2.3 finetuned	2020	Q1	83%	87%	29.14

model 2.3 finetuned	2020	Q2	92%	92%	25.46
model 2.3 finetuned	2020	Q3	88%	91%	22.91
model 2.3 finetuned	2020	Q4	82%	90%	19.74
model 2.3 finetuned	2021	Q1	78%	78%	43.21
model 2.3 finetuned	2021	Q2	63%	64%	35.00
model 2.3 finetuned	2021	Q3	74%	75%	35.75
model 2.3 finetuned	2021	Q4	90%	89%	32.85
model 2.3 finetuned	2022	Q1	80%	81%	34.95
model 2.3 finetuned	2022	Q2	75%	83%	51.80
model 2.3 finetuned	2022	Q3	67%	74%	35.77
model 2.3 finetuned	2022	Q4	91%	91%	40.07
model 2.3 finetuned	2023	Q1	93%	93%	36.47
model 3.3	2019	Q1	64%	68%	11.96
model 3.3	2019	Q2	62%	74%	15.26
model 3.3	2019	Q3	64%	71%	13.39
model 3.3	2019	Q4	83%	85%	15.47
model 3.3	2020	Q1	58%	61%	12.41
model 3.3	2020	Q2	54%	68%	18.29
model 3.3	2020	Q3	47%	64%	27.57
model 3.3	2020	Q4	71%	79%	15.79
model 3.3	2021	Q1	72%	81%	20.06
model 3.3	2021	Q2	70%	73%	19.92
model 3.3	2021	Q3	64%	61%	11.22
model 3.3	2021	Q4	85%	87%	18.13
model 3.3	2022	Q1	62%	68%	26.85
model 3.3	2022	Q2	60%	72%	24.17
model 3.3	2022	Q3	43%	60%	25.75
model 3.3	2022	Q4	83%	88%	27.94
model 3.3	2023	Q1	92%	92%	12.29
model 3.3 finetuned	2019	Q1	64%	68%	30.22
model 3.3 finetuned	2019	Q2	75%	85%	37.68
model 3.3 finetuned	2019	Q3	71%	73%	33.07
model 3.3 finetuned	2019	Q4	78%	84%	38.03
model 3.3 finetuned	2020	Q1	67%	73%	27.64
model 3.3 finetuned	2020	Q2	85%	88%	28.05
model 3.3 finetuned	2020	Q3	88%	94%	35.76
model 3.3 finetuned	2020	Q4	82%	90%	29.75
model 3.3 finetuned	2021	Q1	78%	78%	43.33
model 3.3 finetuned	2021	Q2	65%	66%	45.56
model 3.3 finetuned	2021	Q3	57%	55%	22.46
model 3.3 finetuned	2021	Q4	95%	95%	44.95
model 3.3 finetuned	2022	Q1	76%	80%	41.05

model 3.3 finetuned	2022	Q2	80%	87%	36.91
model 3.3 finetuned	2022	Q3	76%	80%	51.20
model 3.3 finetuned	2022	Q4	91%	89%	65.59
model 3.3 finetuned	2023	Q1	87%	89%	31.91

Test Dataset (Details of Table 8)

Model_Name	Year	Quarter	Accuracy	F1_Score	Run_Time
model 1.3 test dataset	2024	Q1	82%	83%	39.97
model 1.3 test dataset	2023	Q2	71%	75%	34.29
model 1.3 test dataset	2023	Q3	92%	96%	30.33
model 1.3 test dataset	2023	Q4	57%	69%	23.10
model 1.3 finetuned test dataset	2024	Q1	82%	79%	29.68
model 1.3 finetuned test dataset	2023	Q2	71%	69%	22.72
model 1.3 finetuned test dataset	2023	Q3	92%	92%	26.84
model 1.3 finetuned test dataset	2023	Q4	57%	62%	21.20
model 2.3 test dataset	2024	Q1	86%	84%	23.47
model 2.3 test dataset	2023	Q2	84%	82%	25.76
model 2.3 test dataset	2023	Q3	89%	94%	20.82
model 2.3 test dataset	2023	Q4	89%	89%	24.02
model 2.3 finetuned test dataset	2024	Q1	73%	72%	24.89
model 2.3 finetuned test dataset	2023	Q2	71%	76%	17.43
model 2.3 finetuned test dataset	2023	Q3	94%	94%	28.01
model 2.3 finetuned test dataset	2023	Q4	78%	78%	39.47
model 3.3 test dataset	2024	Q1	77%	74%	15.06
model 3.3 test dataset	2023	Q2	74%	74%	14.00
model 3.3 test dataset	2023	Q3	88%	91%	13.11
model 3.3 test dataset	2023	Q4	84%	84%	11.37
model 3.3 finetuned test dataset	2024	Q1	73%	72%	44.97
model 3.3 finetuned test dataset	2023	Q2	74%	72%	46.69
model 3.3 finetuned test dataset	2023	Q3	93%	93%	33.42
model 3.3 finetuned test dataset	2023	Q4	43%	40%	16.85