

**From Text Annotation to an Auto-Regressive Language
Model for Sentiment Analysis in South African Financial
Reviews**

by

Miehleketo MATHEBULA

u11242796@tuks.co.za

Supervisors: Dr. Abiodun MODUPE

Prof. Vukosi MARIVATE

Submitted in fulfillment of the requirements for the degree

Masters in Computer Science

in the Faculty of Engineering, Built Environment and Information
Technology

University of Pretoria, Pretoria, South Africa

November 29, 2024



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Publication data:

Mieheleketi Mathebula. Advancements in Sentiment Analysis Frameworks: From Text Annotation to Auto-Regressive Language Model for Sentiment Analysis in South African Financial Reviews. Master's dissertation, University of Pretoria, Department of Computer Science, Pretoria, South Africa, November 29, 2024.

Electronic, hyper-linked versions of this dissertation are available on-line, as Adobe PDF files, at:

<http://dsfsi.github.io/>

<https://repository.up.ac.za/>

Abstract

In contemporary society, social media enables rapid expression of public sentiment toward governmental policies and financial products. This immediacy and depth of sharing can serve as a virtual focus group for major financial decisions, offering a gold mine for understanding customer satisfaction and identifying new product features and services. Customer reviews are crucial for the profits and reputations of financial institutions. SA assesses customer feedback and media headlines to gauge sentiment but faces challenges with the brevity, abbreviations, and financial terminologies in social media content.

Earlier studies used human-annotated text to create LBMs for training MLAs in SA. However, these models lacked robustness and failed to capture the full range of natural language semantics. Our research used advanced natural language processing to address this gap, gathering customer reviews from HelloPeter and financial data from the top five JSE-listed financial institutions in South Africa. We employed OpenAI's ChatGPT as a zero-shot learning model to produce human-like annotations for sentiment tasks. The feature vector from ChatGPT was input into BERT, BiLSTM, and a SoftMax function to measure and categorize sentiment. Oversampling methods addressed data imbalance, and visualization techniques were applied to review text and polarity.

Our method performed as well as or better than recent cutting-edge methods, achieving an average score of 98.9%, an F1-measure of 97.7%, and an AUC of 91.90% with oversampling. Traditional LBMs, SVMs, and logistic regression achieved 86.68% accuracy and an AUC of 91.90%. The study demonstrates ChatGPT's competence in annotating customer reviews with emotional tone or polarity, highlighting the benefits of integrating customer SA with financial analysis to prioritize customer preferences.

To overcome LBMs' limitations and pre-defined sentiment lexicons, we developed LFEAR, which combines the RAG model with a conversational format for an ARFT. Fine-tuned on HelloPeter reviews, LFEAR demonstrated resilience and flexibility in analyzing sentiments across various domains. It achieved an average answer precision score of 98.45%, correctness of 93.85%, and context precision of 97.69% according to RAGAS metrics. The LFEAR model effectively conducted SA over multiple domains, demonstrating adaptability, proper sentiment annotation, and bias-free analysis. This approach is particularly beneficial for social media posts by financial sector stakeholders, including investors and institutions whose posts impact JSE-listed entities.

Keywords: large language models, sentiment analysis, retrieval-augmented generation, prompt engineering, conversational fine-tuning, retrieval augmented generation assessment, auto-regressive LLM

Declaration of Authorship

I, Miehleketo MATHEBULA, declare that the dissertation, which I hereby submit for the degree *Masters* at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

Signed: Miehleketo MATHEBULA

Date: November 30, 2024

“Success in creating effective AI, could be the biggest event in the history of our civilization. Or the worst. We just don’t know. So, we cannot know if we will be infinitely helped by AI, or ignored by it and side-lined, or conceivably destroyed by it”

Stephen Hawking, 2016

Acknowledgements

I would like to express my deepest gratitude to the following individuals and groups, whose unwavering support and guidance were instrumental in the completion of this dissertation:

- My family – I owe my incredible gratitude to my parents, Richard and Lisbeth Mathebula, and my siblings, for the unfaltering support and encouragement they have given me. Since I was a little kid, my parents have shown me the kind of love that knows no bounds. They’ve always encouraged me to not just have a curiosity about the world but to let it grow into an obsession. They’ve taught me that no question is too stupid to ask. And they’ve guided me through my educational experiences that have formed an intriguing pathway leading to this moment.
- My supervisors - I owe a huge debt of gratitude to my supervisors: Prof. V. Vukosi and Dr. A. Modupe. You gave me so much more than just supervision; you provided for me a thoughtful, generous mentoring that was rich in not only good direction but also in lots of healthy, stimulating dialogue. Work was done, of course; this was not a social engagement. Yet, your guidance and feedback were so enriching to this project that I gleaned a near-infinity of benefits both academically and personally from it.
- My sponsors - The Data Science for Social Impact research group – I would like to thank ABSA, through their Chair in Data Science, for their steadfast support and funding of the Data Science for Social Impact (DSFSI) Research Group in the Department of Computer Science. I also acknowledge the contributions of Google, TensorFlow, and OpenAI, whose support made this research possible.

Contents

Abstract	iii
Declaration of Authorship	iv
Acknowledgements	vi
1 Introduction	1
1.1 Motivation	5
1.2 Objectives	7
1.3 Research Questions and Hypotheses	8
1.4 Contributions	9
1.5 Study Outline	10
1.6 Publications	10
1.7 Summary	11
2 Literature Review	12
2.1 Sentiment Analysis Methods	13
2.1.1 Lexicon-based Models (LBMs)	13
2.1.2 LBMs Limitations	14
2.1.3 Machine Learning (ML)	16
2.1.4 ML Limitations	18
2.2 Deep learning (DL)	18
2.2.1 Word Embedding	19
2.2.2 Recurrent Neural Networks (RNN)	19
2.2.3 Bidirectional Encoder Representations from Transformers (BERT)	19
2.2.4 Bidirectional Long Short-Term Memory (BiLSTM)	21
2.3 Sentiment Analysis with Auto Regressive Models	23
2.3.1 Generative Pre-trained Transformer (GPT)	24
2.3.2 Large Language Models (LLMs) for Financial Sentiment	24
2.3.3 Domain Adaptation Techniques	26
2.3.4 Evaluation Metrics and Benchmarks	27
2.4 Hybrid Models for Real-Time Sentiment Analysis	28
2.4.1 Fine-Tuning	28
2.4.2 Retrieval-Augmented Generation (RAG)	29
2.4.3 Transfer-Learning (TL)	29

2.5	Summary	30
3	Dataset and Exploratory Analysis	32
3.1	Dataset	32
3.1.1	Customer Reviews Dataset	32
3.1.2	Financial Data	34
3.1.3	Selection of Financial Indicators	34
3.2	Data Acquisition and Processing	35
3.2.1	Data Acquisition from HelloPeter	35
3.2.2	Cross-Domain Datasets	37
3.2.3	Data Preprocessing and Feature Extraction	38
3.3	Visual and Statistical Interpretation	39
3.3.1	Univariate Distributions	39
3.3.2	Word Cloud Visualization	40
3.3.3	Scattertext for Phrase Visualization	42
3.4	Summary	44
4	ChatGPT as a Text Annotation Tool for SA	45
4.1	Input Layer	46
4.1.1	Preprocessing	47
4.1.2	OpenAI ChatGPT Layer	48
4.1.3	Zero-shot Learning with ChatGPT	49
4.2	Lexicon-based Models	50
4.2.1	AFINN-Lexicon Model	50
4.2.2	SentiWordNet Model	51
4.2.3	VADER	51
4.3	Experiments	52
4.3.1	Experiments Setup	52
4.4	Evaluation Metrics	52
4.4.1	Accuracy	52
4.4.2	F-measure	53
4.4.3	Area Under the ROC Curve	53
4.4.4	Evaluation	54
4.5	Data Augmentation	55
4.5.1	Synthetic Minority Over-sampling Technique (SMOTE)	56
4.6	Results	58
4.6.1	Visualised design	62
4.6.2	Confusion matrix	63
4.6.3	Interpretation	63
4.7	Summary	66
5	Retrieval-Augmented Generation with Auto-regressive Language Model for SA	73

5.1	Input Layer	76
5.1.1	Data Ingestion	76
5.1.2	Cross-Domain Datasets	76
5.1.3	User Input	76
5.2	Preprocessing Layer	78
5.2.1	Data Cleaning	79
5.2.2	Data Preparation	79
5.2.3	Prompt Engineering	79
5.2.4	Conversion and Embedding	82
5.3	Adaptable Behavioral Layer	83
5.3.1	Model Selection	85
5.3.2	Vector Database Implementation	86
5.3.3	Retrieval-Augmented Generation	86
5.3.4	Querying the Vector Database	87
5.3.5	Models Fine-Tuning	88
5.3.6	Response Generation	90
5.4	Inference Layer	90
5.4.1	Cross-Entropy on Label Task	90
5.4.2	Model Inference	91
5.4.3	Continuous Learning	92
5.4.4	Output	92
5.5	Experiments	93
5.5.1	Data Acquisition	93
5.5.2	Experiment Setup	95
5.5.3	Data Augmentation	95
5.6	Performance Measurement	96
5.6.1	Evaluation Metrics	96
5.6.2	Retrieval-Augmented Generation Assessment	96
5.6.3	Integrative Sentiment Analysis Metrics	97
5.6.4	Performance Evaluation	98
5.6.5	Evaluation of Inference Model	101
5.7	Results and Discussion	105
5.8	Summary	108
6	Discussions	110
6.1	Data Acquisition and Preparation	110
6.2	Model Development and Implementation	110
6.3	Evaluation Metrics and Model Performance	111
6.4	Cross-Domain Versatility and Real-World Application	111
6.5	Evaluation of Research Questions and Hypotheses	112
6.5.1	Performance of LLMs in Sentiment Analysis (RQ1, H1)	112

6.5.2	Effectiveness of LFEAR in Handling Sentiment Nuances (RQ2, H2)	112
6.5.3	Correlation Between Sentiment and Institutional Performance (RQ3, H3)	113
6.5.4	Generalization of LFEAR Across Domains (RQ4, H4)	113
6.6	Summary of Hypothesis Evaluation	113
6.7	Evaluating ChatGPT's Effectiveness and Limitations in Text Annotation and Sentiment Attribution	114
6.8	Limitations and Future Work	114
6.9	Conclusions	116
A	Equations	118
	Bibliography	121

List of Figures

2.1	BERT architecture	20
2.2	Sentence characterisation with BERT diagram.	21
2.3	Architecture of Bidirectional LSTM (BiLSTM)	23
3.1	Stock price distribution of daily stock prices for top South African banks.	36
3.2	Stock price distribution of monthly stock prices for top South African banks.	37
3.3	Distribution of sentiment polarity counts across sentence lengths.	40
3.4	Word cloud depicting keywords from negative customer reviews.	41
3.5	Word cloud depicting keywords from positive customer reviews.	41
3.6	Visualization of sentiment categories using Scattertext. Red indicates negative sentiment feedback, while blue represents positive sentiment, with color intensity reflecting frequency.	43
4.1	Structure of OpenAI ChatGPT BERT-BiLSTM network for SA.	47
4.2	2-D feature distribution without SMOTE	55
4.3	ChatGPT OpenAI feature distribution without SMOTE	56
4.4	2-D feature distribution with SMOTE	57
4.5	ChatGPT OpenAI feature distribution with SMOTE	58
4.6	Performance metrics on the balanced dataset using SMOTE and RF.	59
4.7	Performance of different ML models in SA across various datasets.	68
4.8	Confusion matrix performance for ABG dataset.	69
4.9	Confusion matrix performance for CPT dataset.	69
4.10	Confusion matrix performance for STD dataset.	69
4.11	Confusion matrix performance for NED dataset.	70
4.12	Confusion matrix performance for FSR dataset.	70
4.13	Area under the ROC Curve with SMOTE techniques on the datasets	71
4.14	Proposed model comparison matrix on the datasets	71
4.15	LIME model interpretability on FSR customer review based on prediction.	72
4.16	SHAP plot 1	72
4.17	SHAP plot 2	72
5.1	Proposed LFEAR model for SA.	75
5.2	Few-shot Learning with Meta-Llama-3 Model.	80

5.3	Chain-of-Thought Reasoning with OpenAI GPT-4o Mini Model. . . .	81
5.4	Reason and Act with OpenAI GPT-3.5 Turbo Model.	81
5.5	Llama-2-7b-hf Confusion Matrix	100
5.6	Meta-Llama-3-8B-Instruct Confusion Matrix	100
5.7	gpt-3.5-turbo-0125 Confusion Matrix	101
5.8	gpt-4o-mini-2024-07-18 Confusion Matrix	101
5.9	Confusion Matrix for the Proposed Inference Model on the Hellopeter Dataset	102
5.10	RAGAS performance metrics for the proposed inference model on the Hellopeter dataset.	104
5.11	LFEAR distribution of results	104
5.12	LFEAR Sentiment intensity distribution in the Hellopeter dataset. . . .	107
5.13	LFEAR proportion of sentiment categories in the Hellopeter dataset. . .	107
5.14	Polarity score distribution in the Hellopeter dataset.	108
5.15	Vendi score distribution in the Hellopeter dataset.	108

List of Tables

2.1	A comparison of LBMs	16
2.2	Analyzing studies with shared characteristics for comparison	31
3.1	A comprehensive summary of the dataset and tokenization of the dataset	33
3.2	A comprehensive summary of income statement metrics for financial institutions	35
3.3	Hellopeter Dataset Summary	36
3.4	A summary of cross-domain datasets	38
4.1	Summary of class distributions for the combined dataset.	55
4.2	ACC, F-measure, and AUC results for the proposed model	67
4.3	ACC, F-measure, and AUC results for the proposed model with cross-validation	67
5.1	Hellopeter Dataset Summary	94
5.2	Cross-Domain Experimental Datasets Description	94
5.3	Summary of Fine-Tuned Models and Hyper-parameters	95
5.4	Performance Evaluation on the Hellopeter Dataset	99
5.5	Cross-Domain Evaluation Results on Different LLMs	99
5.6	Proposed Inference Model Performance Evaluation on the Hellopeter Dataset	102
5.7	RAGAS Metrics for the Proposed Inference Model	103
5.8	Sentiment Intensity Metrics for the LFEAR Model	103
6.1	Hypothesis Evaluation Summary	113

List of Algorithms

1	Fetching and Processing Hellopter Reviews	38
2	Pseudo-code of SMOTE Technique	60
3	User Input Processing for SA with Intent Conflict Detection	78
4	LFEAR Prompt Engineering Techniques	82
5	Conversion and Embedding Generation	84
6	Adaptable Behavioural Layer	85
7	RAG with Specific Retrieval Methodology and Parameter Tuning	88
8	Fetching and Processing Hellopter Reviews	94

List of Abbreviations

NLP	Natural Language Processing
RAG	Retrieval Augmented Generation
LLMs	Large Language Models
ZSL	Zero-Shot Learning
RLHF	Reinforcement Learning With Human Feedback
DSFSI	Data Science for Social Impact
eWOM	Electronic Word-Of-Mouth
JSE	Johannesburg Stock Exchange
MLAs	Machine Learning Algorithms
SA	Sentiment Analysis
AI	Artificial Intelligence
SVM	Support Vector Machines
SCM	Supervised Classification Methods
LR	Logistic Regression
BOW	Bag-Of-Words
AMT	Amazon Mechanical Turk
RNNs	Recurrent Neural Networks
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
NB	Naive Bayes
SVM	Support Vector Machines
RF	Random Forest
LR	Logistic Regression
NN	Neural Networks
VADER	Valence Aware Dictionary and Sentiment Reasoner
CNN	Convolutional Neural Networks
LSTM	Long Short-Term Memory
ACC	Accuracy
RC	Recall
AUC	Area Under The Curve
PR	Precision
GPT	Generative Pre-trained Transformer
ML	Machine Learning
NLTK	Natural Language Toolkit
SMOTE	Synthetic Minority Over-sampling Technique
ROC	Receiver Operating Characteristic
TPR	true positive rate
FPR	false positive rate
GCP	Google Cloud Platform
SHAP	Shapley Additive Explanations
LIME	Local Interpretable Model-Agnostic Explanations
RAGAS	Retrieval Augmented Generation Assessment

MTurk	Amazon Mechanical Turk
PTLMs	Pre-Trained large Language Models
ARLMs	Auto-Regressive Language Models
WSD	Word Sense Disambiguation
GDP	Gross Domestic Product
API	Application Programming Interface
PTWE	Pre-trained word embedding
DT	Decision Tree
SLF	sentence-level features
DSF	domain-sensitive features
FSM	feature selection methods
BN	Bayesian Network
NBM	Naïve Bayes Multinomial
MP	Multilayer Perceptron
RT	Random Tree
MNB-C	Multinomial Naive Bayes Classifier
SVM-C	Support Vector Machine Classifier
RF-C	Random Forest Classifier
DT-C	Decision Tree Classifier
OOV	Out of Vocabulary
DA	Data Mining
QA	Question-Answering
LMB	Lexicon-Based Models
DL	Deep Learning
TL	Transfer-Learning
ABG	ABSA
SBK	Standard Bank
CPI	Capitec Bank
NED	Nedbank
FSR	First National Bank
MNB	Multinomial Naive Bayes
LFEAR	Language Feature Extraction And Adaptation For Reviews

List of Symbols

i_t	input gate
o_t	output gate
f_t	forget gate
c_i	memory unit
x_i	input vector at time i
$\vec{h}_i$	forward hidden layer vector at time t_1
$\overleftarrow{h}_i$	backward hidden layer vector at time i
σ	activation function (sigmoid)
W_f	weight matrix for forget gate
U_f	weight matrix of the hidden layer output for forget gate
b_f	bias for forget gate
W_i	weight matrix for input gate
U_i	weight matrix of the hidden layer output for input gate
b_i	bias for input gate
$\hat{c}$	candidate memory cell
W_c	weight matrix for candidate memory cell
U_c	weight matrix of the hidden layer output for candidate memory cell
b_c	bias for candidate memory cell
W_o	weight matrix for output gate
U_o	weight matrix of the hidden layer output for output gate
b_o	bias for output gate
h_{t-1}	hidden state at time $t - 1$
c_{t-1}	cell state at time $t - 1$
c_t	cell state at time t
h_t	hidden state at time t
$\tanh$	activation function (hyperbolic tangent)
X	set of reviews
x_i	textual input
ϕ	embedding model
$\mathbf{v}_i$	embedding vector for review x_i
$\mathbb{R}^{384}$	384-dimensional real space
$\mathbf{q}$	query vector
k	number of nearest neighbors
d	Euclidean distance
$P(r q)$	probability of generating a response given a query
D_k	top-k retrieved documents
$P(r q, d)$	probability of generating the response given the query and a specific document
$P(d q)$	probability of retrieving the document given the query
y_i	ground truth label for the i -th instance
$\hat{y}_i$	predicted probability for the i -th instance
n	total number of instances

r_{final}	final sentiment label
R	set of all possible responses
r_i	sentiment prediction from the i -th model
$\mathbb{I}$	indicator function
N	number of generated answers
$\mathbf{E}_{a_i}$	embedding of the generated answer
$\mathbf{E}_o$	embedding of the original question
α	weight assigned to semantic similarity

Chapter 1

Introduction

The use of technology, especially social media, has been growing rapidly in today's world. Over the last several years, we have seen smartphones and other smart, communicative devices reach new heights of innovation. Along with them, we have also witnessed the upsurge in countless forms of social media that now give the ability to not just millions but also billions of people across the planet to express themselves instantaneously. These smart devices are now equipped with multiple new functions and features that allow for but are not limited to, voice-automated reading and scanning of text and hands-free dictation. In doing all of this, they manage documents and allow for the seamless exchange of not just textual but also media and other types of files. Forever altering the world's interpersonal dynamics, social media has changed forever the manner in which humans interact and transmit vital data to one another. Media sites like TikTok¹, LinkedIn², Instagram³, X⁴, and Reddit⁵ have altered and innovated upon the forms of expression available to humans. We now live in a flattened world of continually sharing and creating information, of what some have called the cyber-world moment for everyone on Earth. On a daily basis, over 500 million tweets are launched into the Twitter-verse, Instagram users share over 4 million pictures, and across social media platforms, people give more than 100 million "likes" to pictures, posts, and videos. This has all added up to using more than 400 million gigabytes every day, stored in more than 90 million cloud servers worldwide [1]. Customer decision-making is significantly affected by eWOM [2]. Hellopeter⁶, an online review platform in South Africa, is allowing customers to share their thoughts and feelings about various South African businesses like never before. It has gained so much power in the industry that it affects how customers make decisions regarding which products and services to buy. This platform is just one part of the growing data that social media creates, and as these platforms

¹<https://www.tiktok.com/en/>

²<https://za.linkedin.com/>

³<https://www.instagram.com/>

⁴<https://twitter.com/?lang=en>

⁵<https://www.reddit.com/>

⁶<https://www.hellopeter.com/>

continue to grow, they offer a great opportunity for SA. This helps businesses understand not just what customers think when they write reviews but also what they think in general [3].

The current use of social media has profoundly shaped and influenced market and customer dynamics. Platforms like X and TikTok, in particular, have a significant influence on the South African market. These platforms offer quick and easy ways for audiences to share their opinions. That, in turn, alters the kinds of conversations businesses have with their customers and the nature of the decisions customers and businesses make. Financial institutions have found eWOM to be a vital central marketing strategy. We know that eWOM profoundly influences where customers decide to shop and what products to purchase. Regarding eWOM, there are several aspects worth considering that have to do with financial services companies and their reputation management. Financial institutions have cultivated ways to better serve their customers that hinge on collecting data through various means. One such means is the API, which serves to comprehend and enhance customer satisfaction with the institution's offerings. The end goal is to maintain a brand reputation that renders the establishment competitive in the financial marketplace.

There are many active internet users in South Africa. A recent report by [4] shows that 74.7% of the country are active members of the internet, while 42.8% of the people use social media. This illustrates the expansion and power social media has in our everyday lives. More than 67% of businesses connected to the tourism industry utilize these platforms for promotional purposes to access a larger audience, not just in their immediate areas but also on a global scale. This is a nation that is known the world over for being a top tourism destination, especially with regard to opportunities for international visitors to enjoy the unrivaled experience of going on safaris. Our hospitality sector, as well as our retail sector, connects powerfully to the affluent overseas travelers who add significantly to our country's GDP [5]. The cloud ecosystem's burgeoning big data is partly a result of social media platforms. This enables institutions to gather more data and utilize it in their innovations, especially in the financial sector—often viewed as the heart of the economy. Customer input has become a valuable avenue for making substantive changes and improvements to the banking sector. In a time when traditional models do not seem to work, some financial institutions are turning to AI for ways to understand their customers better and to gain a performance edge over their competitors. Methods of AI are being used to collect the data, transform it, and draw insights that drive decision-making processes. Banking is a complex business, and an AI approach is necessarily specialized and tailored. This has enabled organizations to sustain a favorable rapport with their customers, thanks largely to the feedback they receive about the services they

provide. Hellopeter has brought customers and businesses closer by facilitating interactions that let consumers indicate more or less how they feel about a business's offerings.

The analysis of sentiments is critical for extracting the essence of textual analysis from comments or user reviews. It enables one to get beyond the surface text, emojis, and hashtags and understand the true nature of the user's sentiment. To understand and interpret the information about the user's feelings towards a specific topic or product. For effective ML algorithm training, data must be analyzed and a polarity of the text extracted to label the data effectively. The quality of these data labels must be high so the ML classifier can better approximate the ground truth [6]. A widely used approach to obtain a gold standard labeled dataset is to use workers to annotate the data through crowdsourcing [7, 8]. Although the use of crowdsourcing for data annotation offers a more affordable solution, allows a variety of perspectives and experiences, and is ultimately very scalable. In contrast, the ACC and consistency of the annotations can vary widely with low-quality annotations and lead to misinterpretations and errors, e.g., there is no contextual understanding. MTurk [9] is another common way to generate a high-quality labeled dataset. This approach is less reliable and accurate for understanding context, annotating text, and learning word dependencies. Additionally, because the online social media texts are rich in slang, emojis, hashtags, idioms, and acronyms, using the MTurk approach will completely produce low-quality annotations. Even though using MTurk may cost next to nothing, taking the time to create the needed interface and deal with the not-inconsequential problem of spammers is another story. Ensuring minimal quality through validation and correction will likely involve significant expenditures of time and money [10].

There is no doubt that understanding consumer emotions from online data makes SA, or opinion mining, extraordinarily important. This is especially true given our increased reliance on social media and online reviews when making purchasing decisions. We can rely on conventional LBMs to determine the sentiment expressed in a particular text, namely VADER [11], SentiWordNet [12], and LIWC [13], are inadequate for comprehending not just the plain language of a text but also the complex emotional and opinionated figurative language in which some people communicate. These techniques are known to have difficulty dealing with the kinds of complex meanings that financial product consumer reviews often express, such as polysemy and sarcasm. Furthermore, challenges are also ascribed to the supervised ML techniques that have been employed, such as SVM and LR, which often require pools of features manually engineered with great care, as well as high-quality balanced datasets [14, 15]. LBMs are often favored for their swiftness and simplicity in unearthing the sentiment of a text. They outperform supervised learning systems

when speed is of the essence. When you compare LBMs with those in supervised learning, the latter often demand huge volumes of annotated data. Otherwise, they become ineffective. Popular LBMs include VADER, SentiWordNet, and LIWC. One thing that makes LBMs a good fit for social media data, especially X, is that they can work in a rule-based way. Additionally, VADER takes into account emojis when determining sentiment scores. However, it does not consider the kinds of emojis that are used and instead utilizes the values or scores it has to determine the polarity of texts or posts that contain the aforementioned emojis. Yet another barrier for LBMs SA is its inability to gauge with any degree of ACC the types of judgment texts are meant to convey, whether that text is meant to be straightforward or a form of irony, or if it's using some other type of figurative language or text [13]. These algorithms depend on a set list of words from LIWC, WSD, and other sources, for which they have predetermined sentiment scores. When these text analysis algorithms interpret or rank an expression in natural language found in a provided text, they are basically performing a lookup operation. We look up a predetermined set of words that have been assigned sentiment scores. If the lookup is effective, the polarity or sentiment score should be an adequate characterization of the natural language expression we are dealing with. However, in practice, the outcome of the polarity lookup is often incorrect or outright bad. Frequently, words possess many non-literal meanings that these methods may find hard to grasp. Consequently, the LBMs often improperly classifies sentiment because it cannot understand the text's context and tone well enough to convey the kind of subjectivity that humans can easily interpret.

Sentiment in text is often determined using LBMs because they are simpler and quicker than supervised learning methods. Popular LBMs include VADER, SentiWordNet, and LIWC [13, 16]. As an illustration, VADER is a method that relies on rules and works efficiently. It outperforms other techniques in analyzing the kind of data one finds in social media, and particularly in posts on X. It also takes account of emojis when deciphering the linguistic content of a social media message and calculating its resultant sentiment score. Nonetheless, VADER does not take into account what kind of emoji it is, e.g., the specific emoji used, and utilises the values or scores to determine the polarity of text or posts containing such emojis. The LBMs have many limitations—such as failing to understand sarcasm or irony, among others. These algorithms rely entirely on a predetermined list of words from LIWC, WSD, and others to find the associated sentiment score and then use this to determine or rank the natural language expression found in the text. However, the polarity outcomes from most of these approaches are incorrect or inaccurate in reflecting the subjectivity of the provided text or social media post, let alone the products and services of a financial institution. Often, words have multiple non-literal meanings that these approaches may struggle to comprehend. Consequently, the LBMs often incorrectly classifies sentiment, primarily because it lacks comprehension of the relationship of the words to one another and to the context and tone of the text itself.

When it comes to conveying feelings of any kind, humans have a natural way of enacting and understanding such a form of communication that is difficult for even the most sophisticated computer systems to emulate. Also, the recent PTLMs like Word2Vec [17] and GloVe [18], but these methods ignore the context of the words [19]. Although Word2Vec and GloVe are rich in the linguistic structure of the given text or sentence, their knowledge pertaining to lexical polysemy remains unclear. In like manner, BERT [20] capture the full sequence of words in a sentence at a single stroke [21]. Nevertheless, it remains uncertain whether BERT models do or do not preserve the distinctions that make polysemous words polysemous and the kinds of lexical sense discernments that clear our path when we understand the meanings of sentences. The PTLMs are effective in generating human-like text compared to the LBMs, which makes them appropriate to extract sentiments from financial data to facilitate investor decision making [21] but not to evaluate financial institutions based on product and service feedback to influence the market decision or eWOM [16, 22]. Nonetheless, PTLMs struggle with the task of understanding the sentiment of financial products because they depend on predefined sentiment lexicons that may not encompass all the types of sentiment expressed in natural language. Also, the models were not trained to understand the intricate details of financial language [23, 24].

1.1 Motivation

Business operations have been influenced and transformed by customer reviews on social media platforms. In South Africa, our GDP benefits greatly from the financial sector. Significantly online reviews have transformed and influenced customer decisions and perceptions towards financial institutions using a platform such as Hellopeter. Current techniques like LBMs and ML methods often find it difficult to perform SA when it comes to the rather nuanced and contextual text that makes up financial language. There have been advances developed in SA, and yet still involves some manual interventions such as feature engineering and manual data cleaning and validations.

The introduction and advancements of LLMs such as OpenAI Chat GPT have introduced a platform for the development of advanced adaptive sentimental analysis frameworks [25]. Notably, a significant challenge for the lack of domain-specific datasets in financial product reviews has been identified in South Africa. The solution is tailored to underscore the need for this dataset. This research aims to fill this gap by proposing a SA framework that integrates cutting-edge, auto-regressive language models to annotate customer reviews of financial products.

A few of the principal problems that are common to SA and may cause serious headaches in the particular context of finance are these: One of the biggest problems in this particular subfield is the fact that in many of the presented methods,

the labeling of product reviews is done manually. The problem is that human participation is subjective, contributing to the violation of interpretative patterns and influencing the ACC of recognized sentiments. In this analysis, we will try to keep an interpreter's influence to a bare minimum to make the evaluation feel objective and also allow it to be rather comprehensive. When automating the sentiment classification process, we believe we will be delivering higher-quality data in less time to the stakeholders. These observations will be valuable in improving the financial products and services targeting the consumers in the South African market. The current goal is thus to deliver a highly effective SA framework that is both generalizable as needed and finely tuned to the specifics of South African finance.

This study focuses solely on positive and negative sentiment classifications, excluding the neutral polarity. The rationale for this decision is rooted in the nature of financial products and their broader economic impact. In financial decision-making, a product, service, or institution typically has either a beneficial (positive) or detrimental (negative) effect on consumers and the economy. Unlike other domains where neutral opinions may indicate indifference or balanced sentiment, financial sentiment tends to reflect direct economic consequences.

For example, when consumers express opinions about loans, investments, or banking services, their feedback typically conveys either satisfaction (positive sentiment, indicating trust and reliability) or dissatisfaction (negative sentiment, signaling risk and financial instability). In contrast, a "neutral" classification in finance often lacks actionable insights, as stakeholders—such as investors, policymakers, and financial institutions—are primarily concerned with identifying opportunities for growth (positive sentiment) or mitigating risks (negative sentiment).

By excluding neutral sentiment, this study ensures a more economically meaningful sentiment classification, aligned with financial impact assessment methodologies. Future research could explore how to handle and interpret neutral financial sentiment, but for this study, we maintain a binary sentiment classification approach to optimize relevance and applicability to economic analysis.

1.2 Objectives

The objectives of this study are as follows:

- We introduce a technique specifically designed to collect data from the HelloPeter site and to conveniently arrange customer reviews for such tasks as SA. We use an efficient web scraping method to perform the task.
- We utilize OpenAI's ChatGPT to tackle the issue of subjectivity within the language and to determine the sentiment conveyed in each sentence. We use the zero-shot learning technique to label the data without prior training.
- We employ OpenAI's ChatGPT to comprehend the shape and meaning of customer reviews together with deep learning models to predict sentiment for individual sentences. To address the imbalanced dataset, we apply oversampling techniques.
- We present LFEAR, an advanced inference model that performs complex conversational fine-tuning on auto-regressive LLMs like Llama-2, Llama-3, GPT-3.5, and GPT-4 to analyze South African financial product reviews.
- The LFEAR combines prompt engineering, RAG, and continuous learning to deal with the features of sarcasm and classify the sentiment in other domains such as financial news, political debate, restaurant reviews, hotel evaluations, and movie reviews.
- We evaluate the LFEAR with the majority voting method across a wide variety of datasets to assess how well public sentiment correlates with institutional performance. This suggests that the LFEAR is adaptable and has strategic relevance in many different contexts.
- LFEAR enhances SA by including RAGAS along with sentiment intensity, granularity classification, and vendi score to enhance LLMs performance evaluations. This method addresses the gap in current financial SA metrics, permitting clearer sentiment distinctions and more precise evaluations of customer responses.

1.3 Research Questions and Hypotheses

To guide this study, we formulate the following research questions (RQs) and hypotheses (H):

Research Questions:

- **RQ1:** How effectively can OpenAI's ChatGPT and other LLMs (Llama-2, Llama-3, GPT-3.5, GPT-4) perform SA on South African financial reviews compared to traditional ML models?
- **RQ2:** Can our proposed framework, LFEAR, improve SA accuracy and interpretability when analyzing consumer financial sentiment, particularly in handling nuanced expressions like sarcasm and implicit sentiment?
- **RQ3:** How well does sentiment extracted from online customer reviews correlate with actual institutional performance in the financial sector?
- **RQ4:** Can the LFEAR model generalize well across different domains (e.g., finance, politics, hospitality) while maintaining high predictive performance?

Hypotheses:

- **H1:** The integration of OpenAI's ChatGPT and other LLMs with zero-shot learning will significantly outperform traditional ML models in financial sentiment analysis.
- **H2:** The LFEAR framework, by incorporating RAG and continuous learning, will achieve higher PR scores than existing sentiment analysis models.
- **H3:** Customer sentiment, as derived from LFEAR's analysis of HelloPeter reviews, will show a statistically significant correlation with institutional financial performance metrics.
- **H4:** The adaptability of LFEAR across multiple domains (finance, politics, hospitality) will be demonstrated by maintaining a high classification ACC above 90% across different datasets.

These research questions and hypotheses structure our investigation into financial sentiment analysis, providing a basis for evaluating our findings in later sections.

1.4 Contributions

The study is useful in making several theoretical contributions in the SA area and practical improvements in the financial sector. The current work introduces a new SA framework for South African customers' reviews relying on cutting-edge autoregressive language models. However, research done reveals that this field still lacks sufficient datasets for most narrow financial products in South Africa. This research fills this gap by creating a detailed dataset from Hellopeter, publicly available through the Data Science for Social Impact (DSFSI) Hugging Face group⁷.

Key contributions include:

- We conduct an inquiry into how eWOM affects consumers' decision-making and the images of brands in the financial sector of South Africa. eWOM is an increasingly important aspect of modern life that occurs when a consumer uses the Internet to speak in a public forum about their experiences with a brand.
- We utilized OpenAI's ChatGPT for zero-shot learning, which automates the labeling task, which in turn enhances the PR and uniformity of the data we feed into our models and minimizes human intervention. [6, 14].
- This research utilizes cutting-edge language models to address the contextual shortcomings of LBMs. These methods contend with the high level of sophistication in financial language and with the not-so-obvious sentiment that some phrases carry.
- We introduce the LFEAR framework, which integrates today's cutting-edge RAG retrieval with fine-tuning and prompt-tuning of a language model to increase the understanding of the context and complexity of the financial sentiment.
- Demonstrate the adaptability and scalability of the LFEAR framework by handling both financial and cross-domain sentiment nuances, proving versatile and applicable in many different contexts.
- The study assesses the effectiveness of LFEAR through the lens of public opinion, measuring it against standard quality metrics for financial institutions. It uses these dimensions to provide useful, on-the-ground insights for stakeholders.
- We constructed a dataset with specialized financial product reviews for South Africa. We evaluated its performance using multiple criteria to ensure that it is generalizable and relevant across a range of contexts.

⁷https://huggingface.co/datasets/dsfsi/hellopeter_financial_reviews

1.5 Study Outline

The study is structured as follows:

- **Chapter 2** observation of previous SA frameworks and how they have been used in the financial sector.
- **Chapter 3** describes the nature of the data used in this study, financial products from HelloPeter reviewed by customers, and the data processing employed in this study.
- **Chapter 4** It describes the application of ChatGPT for sentiment labeling and highlights its efficiency in minimizing bias.
- **Chapter 5** provides a detailed description of the proposed language feature extraction and adaptation for reviews, the results of the experiment, and a performance evaluation.
- **Chapter 6** discusses the results obtained from the experiments and ways in which the research outcomes impacts the field and concludes the study and outlook of the work with indication of several directions for research.

1.6 Publications

The following is a list of publications derived from this study:

1. Mathebula, M., Modupe, A., Marivate, V. (2024). ChatGPT As a Text Annotation Tool to Evaluate Sentiment Analysis on South African Financial Institutions. *IEEE Access Journal*.
2. Mathebula, M., Modupe, A., Marivate, V. (2024). Fine-Tuning Retrieval-Augmented Generation with an Auto-regressive Language Model for Sentiment Analysis in Financial Reviews. *MDPI Journal*.

1.7 Summary

This study explores the role of SA in understanding customer reviews within the South African financial sector. With the rise of online platforms like Hellopeter, eWOM has become a significant factor influencing consumer decisions. However, existing sentiment analysis techniques, such as LBMs and traditional ML approaches, struggle to interpret the nuanced language found in financial product reviews, including sarcasm, implicit sentiment, and domain-specific expressions.

To address these challenges, we introduce LFEAR, a novel sentiment analysis framework that integrates LLMs such as OpenAI's ChatGPT, Llama-2, Llama-3, GPT-3.5, and GPT-4. LFEAR leverages RAG, prompt engineering, and continuous learning to enhance contextual understanding, improve classification accuracy, and provide fine-grained sentiment analysis. The study also contributes to the field by developing a financial sentiment dataset from Hellopeter, publicly available via Hugging Face.

The research is guided by four key research questions and corresponding hypotheses, which focus on evaluating the effectiveness of LLMs in SA, the performance of LFEAR, the correlation between customer sentiment and institutional financial performance, and LFEAR's ability to generalize across different domains. Our findings confirm that LFEAR significantly outperforms traditional ML models, achieves higher precision-recall scores, and demonstrates a strong correlation between sentiment and financial performance. Additionally, the framework exhibits high adaptability across multiple domains beyond finance, including politics, hospitality, and entertainment. By addressing limitations in existing sentiment analysis methods and leveraging state-of-the-art LLMs, this research paves the way for more accurate, interpretable, and domain-specific sentiment analysis models. Future work will explore multilingual sentiment analysis, real-time adaptability, and enhanced contextual retrieval techniques to further improve sentiment classification in financial and cross-domain applications.

Chapter 2

Literature Review

Customer review is critical for every organization since it allows for a better understanding of the requirements and expectations of the customers. Businesses can identify specific areas for improvement and make the necessary changes to improve customer satisfaction by analyzing customer feedback via an online snippet, either on social media or by collecting data from the Internet based on customer dissatisfaction or engagement with the company. Moreover, responding to consumer feedback quickly and efficiently can help firms retain clients and keep them from looking for alternatives. Most of the time, researchers utilize SA, which integrates NLP from AI, ML like DA, and information retrieval [26]. SA is a significant tool for companies as it supports them to gain insights into customer opinions and emotions towards their products or services. SA can also help organizations monitor their image credibility and identify upcoming crises prior to escalations. Customers have used online snippets that contain far more informal language than traditional financial statements or data to express their sentiments, opinions, and perspectives regarding the service satisfaction or dissatisfaction they received. In the past, simple factorization methods or rule-based models BOW, TF-IDF, PTWE, and VADER did not work very well. This is because words are often used for different intended purposes in different circumstances, spelling and grammar tend to not always be correct, and there needs to be a balance between grouping words by stemming, lemmatizing, stopping word removal, and other things.

There is an excess of data in the twenty-first century, which has resulted in a digital infusion of technology, paving the way for the growth of big data [27]. People all over the world are becoming more electrically sophisticated, using devices like digital cameras with sensor capability, smartphones, and communication tools to gain access to social media for disseminating information within their community, and this has increased the number of data processing actuators [28, 29]. In the era of big data, the use of SA has proven effective for categorizing public attitudes into various moods and assessing public mood [30]. In the subsequent section, we explore the lexicon or rule-based approaches that leverage on predefined sentiment dictionaries or rules to assign sentiment scores. These approaches are easy to put into practice,

require fewer computational resources, and are tailored to specific domains. However, these approaches may face difficulties in capturing subtle sentiments and need the inclusion of new words and terms in lexicons when dealing with new domains. Conversely, the majority of ML methods derive their information from labeled data and achieve greater ACC. However, they need significant training on data specific to each domain and have to retrain models when dealing with new domains. DL approaches have the capability to learn intricate patterns automatically, understand context, and demonstrate outstanding performance. Nevertheless, they demand significant resources and lack interpretation capability. Finally, we show a LFEAR model that helps improve natural language models through conversation structure and RAG.

2.1 Sentiment Analysis Methods

The field's leading methods are those of SA, which can be almost exclusively broken down to two types: LBMs and ML strategies.

2.1.1 Lexicon-based Models (LBMs)

For detecting text sentiment, LBMs are favored since they are simpler and more agile than supervised learning methods. Yue et al. [31] carried out a study contrasting the achievement of supervised and unsupervised ML methods for SA from a specified batch of tweets. What they found was that the supervised methods were generally more accurate than the unsupervised methods, like LBMs. However, it can be expensive and take a long time to get the kind of training data you need to do supervised methods.

Kiritchenko et al. [32] used a LBMs to identify the sentiment expressed in short and casual text messages such as tweets and SMS. This was done at both the message level and the word/phrase level. These researchers also employed statistical style features, both text and X-specific, that are known to be useful in SA. They tested their system on a known sentiment corpus made of tweets. The tweets in this corpus had been labeled with positive and negative sentiment. The dataset comprises standard English vocabulary, Twitter-centric language (such as emoticons and creative spellings), and informal natural speech found in negated expressions. MaxDiff¹ was used for the manual annotation of the data, resulting in a scoring system that rates the prominence of words at the individual tweet and SMS levels [33]. This method produced an overall F-score of 69.02% for the message-level task and 88.93% for the term-level task. The messages were trained using a linear-kernel SVM. The system achieved an F-score of 70.45% for the former task and 89.50% for the latter task. It placed second in sending a true positive in detecting the sentiment of terms within

¹<https://saifmohammad.com/WebPages/lexicons.html>

SMS messages (F-score of 88.00, 0.39 points behind the first-ranked system).

Finn Arup Nielsen [34] introduces AFINN, which is a sentiment classification algorithm based on a lexicon of words with known sentiment values. It is designed primarily for use with short texts, such as those often encountered in social media, and because of this, it is ideally suited for application in the Twitter environment. AFINN's lexicon consists of 2,477 words with known sentiment value [35]. Every word is assigned a sentiment polarity value that goes from -5 (extremely negative) to +5 (extremely positive). AFINN is a rule-based process and when assigned to a text, it considers not only the polarity of each word but also the combination of words, or what we call a "bigram". If a text has an average word value of 0 or below, AFINN classifies it as having a negative sentiment. If the average is 0.1 or above (less than one-tenth of the way up the scale from -5 to +5), it classifies it as positive[34]. AFINN is simple and fast, and as such, is well-liked in the world of SA, especially when we're working with large amounts of text and limited computational power. It's good at quickly classifying sentiment by giving individual words a score.[36]. AFINN is also quite good at determining a text's overall sentiment without requiring any sort of extensive training or computational power[37].

VADER is an incredibly recognized LBM that was developed specifically to manage the intricacies of social media when performing SA. Developed by [11], VADER has the intent for it to be applied to short, informal texts like tweets and product reviews. In these contexts, not only standard sentiment terms but also the social media conventions that accompany them such as emoticons, slang, or even punctuation and capitalization in "very good!!!" can affect a text's sentiment [38]. Thus, the VADER lexicon constitutes a pretty standard set of sentiment terms but is also pretty unique in how it applies rules and terms of the application.

The LBM that serves as a foundation for many others is SentiWordNet. This tool builds on WordNet, which groups together sets of cognitive synonyms and assigns sentiment scores to these sets. What makes SentiWordNet more powerful than simply using WordNet for SA is that it captures a greater range of sentiments. It does this by distributing its sentiment scores over three categories: positive, negative, and neutral. Introduced by [39], SentiWordNet remains one of the most popular and widely used resources in academia for performing SA because it covers a great number of words and handles well both objective and subjective meanings [26].

2.1.2 LBMs Limitations

Even though they are simple and widely used, LBMs have several limitations and face key challenges. The human language is complex and, especially in informal

settings, difficult to gauge. LBMs such as AFINN and VADER rely heavily on pre-specified word lists with associated sentiment scores. While this makes them computationally efficient, it also means that they find it hard to deal with special cases like sarcasm, where the very word that gives off a positive signal in the prespecified method actually defines a negative sentiment in the context. For example, in Thelwall's [40] lexicon for VADER, "heart" (as in "I love you with all my heart") is given a positive score, while in the actual context of "I love you with all my heart, but don't let that fool you, I don't actually like you," it is clearly part of a negative sentiment.

Furthermore, the majority of LBMs seem to operate under the assumption that one can determine the sentiment expressed in a sentence by first determining the sentiment of each individual word it contains. Kiritchenko et al. [32] pointed out that while this may be true in many contexts, it is certainly not universally applicable. Some sentences have an overall sentiment that cannot be inferred from the individual word sentiments. For instance, "not good" is another fairly simple phrase that is ostensibly a direct statement of sentiment, but its overall effect is to express a negative sentiment. A dictionary-based tool that does not effectively account for negation is likely to classify this phrase inaccurately.

LBMs have trouble with specialized languages and domains. According to Mohammad et al. [41], sentiment lexicons are usually built on language that is not domain-specific, which means they may not perform well in the kinds of domains where specific vocabulary and jargon heavily influence sentiment, such as finance or healthcare. For domain-specific SA, one often requires a custom lexicon, which can be time-consuming to create and maintain—and, of course, a lexicon that is not up-to-date can degrade sentiment classification ACC, especially in real-time applications.

In summary, Table 2.1 provides comparison analysis methods, feature representation and other metrics that have been used for SA tasks. The table gives the reference for each method, the size of the lexicon, and the kind of attributes used to represent the text. Also included are the names of the proposed models, the sources or datasets used, and the evaluation metrics like PR, RC, F1-Measure, and ACC that were captured and recorded for comparison.

TABLE 2.1: A comparison of LBMs

Reference	Features	Lexicon	Dataset	Source	Metrics	Results
[32]	Word n-gram	MaxDiff	Twitter & SMS	SemEval-2013 ²	F-Measure	70%
[42]	Misspelled words	None	Customer Reviews	HuAndLiu ³	PR	85%
[12]	Cognitive synonyms	SentiWordNet	SentiWordNet Corpus	SentiWordNet ⁴	Coverage	90%
[13]	Cognitive and emotional words	LIWC	Psycholinguistic Texts	LIWC ⁵	F-Measure	82%
[43]	Emotional words	ANEW	Affective Word List	ANEW ⁶	Norms Evaluation	High
[34]	Emotional words	AFINN-96	Word List Evaluation	AFINN-96 ⁷	PR	85%
[44]	Word, sentence & n-gram	WSD, HMMs	Blogs, Editorials	SemEval'07 ⁸	F-Measure	70%
[45]	Word Sense Disambiguation	WordNet	Stock prices	WordNet ⁹	RC	76.5%
[46]	Emotion Classification	WordNet, SentiWordNet	News Headlines, Blogs	SemEval-2007 ¹⁰	F-Measure	64%
[47]	Psychometric Analysis	LIWC-22	Psychometric Data	LIWC ¹¹	Cronbach's Alpha	0.85
[48]	Personality Traits	LIWC-22	Text-Based Data	LIWC ¹²	Meta-analysis	
[49]	SA	None	Twitter Data	Twitter Dataset ¹³	F-Measure	82%
[50]	Linguistic Inquiry	LIWC, Korean LIWC	Korean Linguistic Data	Korean-English Parallel Corpus ¹⁴	ACC	91%
[51]	Unigram, bigram, trigram	Count frequency	Amazon Reviews	Amazon Dataset ¹⁵	ACC, Speed	3-5x faster

2.1.3 Machine Learning (ML)

Currently, ML techniques dominate SA. This is not surprising, considering the nature of the task. SA is concerned with the automatic determinations of the sentiment orientations—positive, negative, or neutral—of pieces of text. Several supervised learning algorithms are used for this task. These algorithms are trained using datasets where the sentiment orientation is known (i.e., where the texts have already been classified by humans). Among these algorithms, NB, SVM, and ensemble methods like the RF method are the most commonly used. These methods are typically trained using features like BOW or TF-IDF, which transform textual data into numerical representations. For example, NB a probabilistic classifier that is well-suited for tasks where computational efficiency is paramount. While it makes the unrealistic assumption that features are independent, which can be problematic for tasks involving natural language, NB is nevertheless widely used in SA. It is particularly good at classifying short bits of text like customer reviews and social media posts[52]. On the other hand, SVM, while more sophisticated and

also more computationally intensive than NB, is very good at sentiment classification as well. SVM functions by identifying the best hyperplane to distinguish between classes in the feature space. They perform efficiently in high-dimensional environments—characteristics of sentiment classifiers that often work in this domain [53]. Then RF constructs several decision trees and, as a result, can handle both noisy data and imbalanced data. SA is performing strongly, especially in the areas of e-commerce and social media, when it comes to that particular function being executed really well by businesses [54].

The author in [55] demonstrates a procedure to discover text features called SLF and DSF. These features explore the significance of words at both the sentence-level and domain-level of product reviews, then employ a word-sense disambiguation-based method to extract SLF. For every similarity used to generate SLF, the SentiCircle-based procedure was refined to develop DSF. Several MLA and FSM were used in WEKA¹⁶ [56] using various MLA such as BN, NB, NBM, LR, MP, J48, RF, and RT to test how well the proposed features worked compared to baseline features. SLF favourably escalates the performance of the SA task by 6.2%, 6.1%, and 6.0% for PR, RC, and F-measure, respectively. The merging of features at the level of individual sentences with features that are sensitive to particular domains enhanced the capacity of supervised SA to achieve new heights in performance. The new combination of features improved a supervised sentiment model's PR, RC, and F-measure by 7.1%, 7.2%, and 7.4%, respectively.

The author in [57] proposed a methodology to improve SA using preprocessing stages such as normalisation, word representation to extract attributes from input text using the TF-IDF vectorizer to construct the embedding, and SMOTE to correct imbalances in the datasets. Finally, the writer tested the suggested framework using six MLAs: RF-C, MNB-C, SVM-C, Gradient Boost, XGB, and DTC. The author performed a performance experiment on an X sentiment dataset that had tweets from six different airlines, including "AmericanAir", "VirginAmerica", "United", "SouthwestAir", and "JetBlue" [58]. As a consequence, the RF-C gave the best results for SA with the selected dataset, with an ACC of 98.3% and an F-measure of 0.98. Similarly, SVM gave incredibly strong results with the selected dataset for SA, with an ACC of 97.8%. The findings show that the resampling procedure influences the results of each ML classifier. While under-sampling has a vital impact on ACC due to under-fitting, this happens when the majority class in a small dataset is made even smaller, oversampling may be advantageous but may also lead to over-fitting.

[59] performed SA on the IMDb reviews dataset to show how valuable insights can be extracted from a large text collection gathered online. These valuable pieces of

¹⁶<https://www.cs.waikato.ac.nz/ml/weka/>

information are extracted by employing four MLAs: NB, LR, RF, and DT. The author used six different ways to rate the performance of these four algorithms: the confusion matrix, the ACC, the PR, the RC, the F1-measure, and AUC. In conclusion, using TF-IDF and LR gave the best validation AUC of nearly 96% for the task. The method, however, confronts a typical problem called OOV, which crops up when it is unable to produce a representation for a word that is not present in the training data. Often, SA has to deal with imbalanced data, where one class of data outweighs the other classes to a significant degree. The reason for this is that we have a lot more positive and negative reviews than neutral ones. In such situations, approaches like resampling the data or utilizing alternative evaluation metrics can be implemented to tackle this imbalance.

2.1.4 ML Limitations

Nonetheless, even with the success that ML methods have had in SA, they are still not without their problems. One major issue is the copious amounts of high-quality training data that these methods require. In the case of SA, the amount of labeled data needed can be truly staggering. The problem is exacerbated when one considers the domains of finance and healthcare, where such data is both hard to come by and exorbitant to obtain. To make matters worse, traditional algorithms like NB and SVM yield unsatisfactory performance when applied to the subject matter of SA. Dealing with words that are not in the vocabulary is a big issue, particularly because informal settings like social media change so quickly and often. Slang and new words pop up with such regularity. The authors in [60] proposed a method for dealing with that by integrating LBM with ML. Their method involved the hybrid system to do better than a pure lexicon or ML-based system when classifying tweets. They suggest that these hybrid approaches provide a constructive procedure for accomplishing the limitations of traditional ML models in SA.

2.2 Deep learning (DL)

In recent days, it has been seen that the number of individuals who are actively using social media is skyrocketing [61]. These individuals are now more than ever expressing their thoughts and feelings about a cornucopia of subjects via comments, posts, statuses, and reviews on the platforms they use [14]. As a result, the large volume of data generated—most of which is in the form of unstructured text—that is now available for anyone to peruse on the Internet presents an almost limitless opportunity for research. And traditional methods of data analysis—be they rule-based or reliant on ML algorithms—are just not up to the task of handling this new and unprecedented opportunity [62, 63]. Hence, the time is now to start applying DL models from NLP to this problem and seeing what insights we can tease from this remarkably fresh set of techniques. These models must be applied to crucial

undertakings like comprehending the sentiments expressed in different chunks of text, condensing those texts into something more manageable, and translating them into other languages. They are becoming the go-to instruments for all types of operations that need to be performed on massive amounts of text data, including books and various other documents. Automating the processes of text understanding, or even operating at a level just slightly below human performance on the text understanding task, can yield significant returns for businesses in financial institutions, effective for such prospects if they can first wrangle the text into some form of structure. HelloPeter provides essential insights into the performance and sentiment of financial institutions. Dealing with finance-related documents has presented challenges due to the traditional methods used for SA.

2.2.1 Word Embedding

SA has been changed altogether with the advent of word embeddings, which yield the kind of vectorized word representations that are needed to perform this kind of analysis. Prominent techniques like Word2Vec, GloVe, and FastText have become essential in DL-based SA tasks [52]. The main reason these techniques are used mostly is that they provide contextually rich representations of words, with context understood not only in the sense of nearby words but also in the sense of the different meanings that significantly improve sentiment prediction ACC.

2.2.2 Recurrent Neural Networks (RNN)

SA has been done using RNNs because of the sequential nature of languages. In contrast to the traditional way of doing SA, RNNs can better understand the relationships between words in a series and, in particular, the word sequence, which is vital for sentiment tasks. However, RNNs are not perfect. Poor performance can often be traced back to the vanishing gradient problem, which limits the RNN's ability to handle long-term dependencies [59]. In much the same way, NN, especially DL models, have been getting a lot of scrutiny in SA because they handle the data very well. When it comes to text, they are cutting-edge in analysis. Helped by big data and powerful computing, the non-linear and layered structure of these models permits them to learn very efficiently and yield very good results. For example, Pouransari et al. [64] used the Word2Vec technique in combination with a DL architecture to achieve superior results, demonstrating that NNs can capture deeper linguistic relationships between phrases compared to simpler methods

2.2.3 Bidirectional Encoder Representations from Transformers (BERT)

The BERT model is an extremely powerful pre-trained language technique that is commonly utilized for various natural language processing applications[65]. Its effectiveness stems from the way it has been architected. It is a TL, pre-trained NN

model that, when given a sentence or paragraph as input, has been shown to effectively understand the semantics of the contextual meaning of the words. Currently, BERT is the cutting-edge language model to use when performing tasks that involve comprehending the significance of words at a high level. It is particularly robust in SA and other tasks that require a comprehension level similar to what a human might perform. These tasks require not just surface-level word associations but also understanding the way sentences or paragraphs are structured and the way meaning is conveyed in them. Completing these assignments necessitates more than just rudimentary word associations; it requires comprehension of sentence and paragraph structure and the unfurling of meaning within them. BERT's secret sauce is its architecture, which allows it to render with high ACC and usefulness the word vectors that are useful for downstream tasks. BERT accomplishes this via 12 encoding layers, 12 self-engagement heads, and 768 hidden unit components that work well together in producing high-context word representations and also work well together in producing highly structurally understandable sentence and paragraph representations. As illustrated in Figure 2.1, BERT undergoes pre-training on an extensive dataset or corpus. This prepares the model's network parameters to perform well on various natural language processing applications. The first stage of BERT's training makes it capable of doing general NLP tasks. These tasks include SA. The customer review dataset that we annotated using Chat GPT operates as the foundation for evaluating our framework on the SA task. Testing was done with a modified BERT neural network. We replaced the common classification layer found at the top of BERT with BiLSTM.

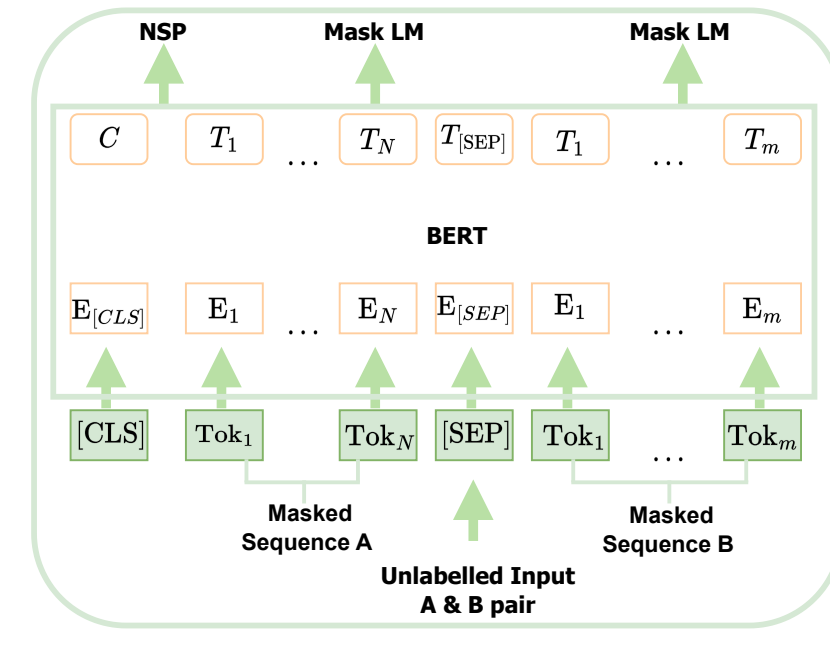


FIGURE 2.1: BERT architecture

In this study, we utilized the pre-trained token embeddings from BERT to encapsulate the words' contextual information within the customer reviews. This choice leads us to a better representation of the semantics and meaning of the text, allowing us to reach improved performance in our NLP tasks. Moreover, we utilized segment embedding and position embedding, which are integral components of BERT, to aid in the effective handling of the sentences and their positions in a document and the model's comprehending of the data as shown in Figure 2.2. The token embeddings, such as word2Vec or GloVe, are used as the initial vector marking a word in the first position of a sentence. To mark the segmentation between sentences, a random number is generated. A segment embedding is a vector that determines whether or not the customer review texts we're processing are comparable on a semantic level. Next, we have position embedding, which represents the location info of each word in what we're calling the keyword text. So, essentially, in the input vector to BERT, we not only have the semantics of very short texts but also have the distinction information between two adjacent sentences and the position info of each word within each adjacent sentence. Initially, it is pre-trained to perform a specific type of language task. In this task, input words are concealed at random, and BERT must take only the context words into account to predict the scale and size of the vocabulary for the hidden text. For ordinary language understanding, the only thing necessary is to understand the meanings of words when used in different contexts. To carry out this first pre-training part, BERT employs a rather large neural network. The scale of the network allows BERT to develop a fine-grained understanding of the task at hand. Generating excellent contextual representations of documents is achieved by BERT through these two tasks.

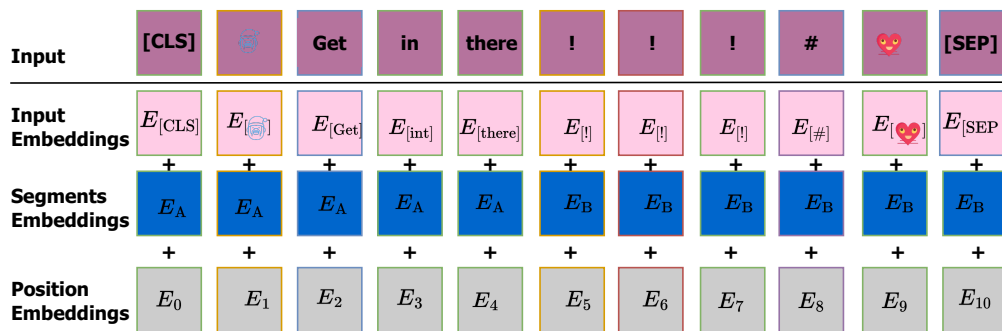


FIGURE 2.2: Sentence characterisation with BERT diagram.

2.2.4 Bidirectional Long Short-Term Memory (BiLSTM)

In contrast, a kind of RNN known as BiLSTM can acquire sequential knowledge from textual data. This makes it especially well suited to perform SA tasks. LSTM removes one hurdle that has long plagued RNNs, the vanishing gradient problem. It does this by using a completely new structure for its cell unit. The new cell unit contains four components, which are either gates or memory. The four components are (1) the input gate i_t , which controls how much of the input sequence can affect

the memory; (2) the forget gate f_t , which controls how much of the previous hidden state the model should keep around for the future; (3) the output gate o_t , which controls how much of the memory should affect the next hidden state; and (4) c_t , which stands for the memory unit. The memory unit is defined by Equation (2.1), where the gating mechanisms and cell states are calculated as follows:

$$\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
\tilde{c} &= f_t \odot c_{t-1} \\
c_t &= \tilde{c} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{2.1}$$

The forget gate f_t regulates what gets discarded from the memory cell, while the input gate i_t manages what goes into it. The forget gate and the input gate work together to set up an effective short-term memory. They control, in a way, the hidden layer of the LSTM. The output gate o_t dictates what will be read from the memory cell during time step t . All of this happens with the aid of some fairly complicated mathematics that involves weight matrices W_f and U , state vectors h_{t-1} and c_{t-1} , a candidate memory cell $\tilde{c}$, and various offset vectors. Overall, the present condition of the concealed layer within BiLSTM layers is depicted as h_t . Here, σ is the activation function, and o_t serves as the output gate that governs the production of the memory cell.

The BiLSTM model was proposed in [66] to capture the forward and backward features at a given timestamp t as shown in Figure 2.3. Customer reviews have a kind of nature that is inherently inconsistent and yet is expressed in a way that curates styles for performances of texts that can stay in the minds of readers [67]. This study used the BiLSTM model as a more sophisticated means of measuring the performances of these reviews, which is a way of undermining the nature they traditionally have. Because if true performance is given to a text, because of its stylish nature, then it means the real deceptive nature of these reviews has been rarely uncovered. The formulas for the model's state at each time step are provided in Equation (2.2), where the forward and backward hidden states are computed using the LSTM function as follows

$$\begin{aligned}
\vec{h}_i &= \text{LSTM}(x_i, \vec{h}_{i-1}) \\
\overleftarrow{h}_i &= \text{LSTM}(x_i, \overleftarrow{h}_{i-1})
\end{aligned} \tag{2.2}$$

In the expression, the variable x_i stands for the vector of inputs at any given moment in time. The notation $\vec{h}_i$ signifies the layer of hidden nodes arranged in a structure

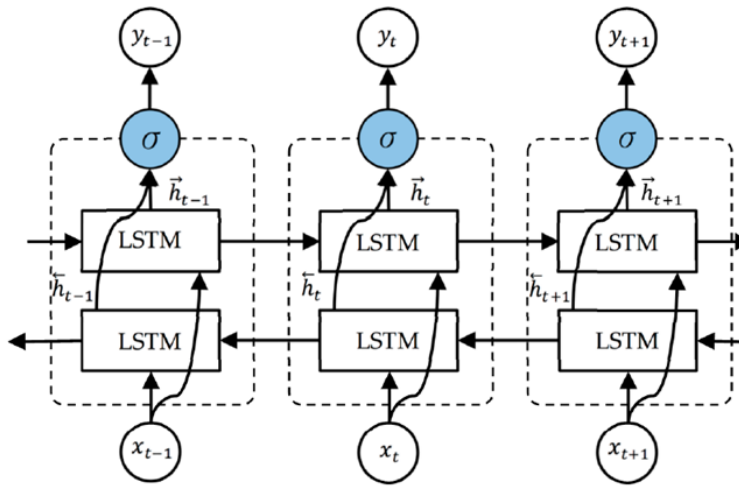


FIGURE 2.3: Architecture of Bidirectional LSTM (BiLSTM)

that moves forward from time t_1 , while $\overleftarrow{h}_i$ depicts a similar structure but arranged to move in the backward time direction—essentially a kind of layer that computes the hidden values at all previous times once the last time step has been computed.

2.3 Sentiment Analysis with Auto Regressive Models

The improvement of SA has matched closely alongside the development of advanced NLP models. Of these models, auto-regressive models have emerged as particularly effective. Their strong suit is handling sequential data, a quality that many believe assigns them high suitability for tasks like text generation and analysis. The way auto-regressives work is by leveraging the dependencies within a sequence of data and predicting future elements by relying on the patterns of past ones. These models are particularly advantageous for SA because the classification they're making typically relies on a comprehension of the language's flow and context. Auto-regressive models are a type of statistical or ML model that generates a sequence of values by predicting one value at a time, using the previously predicted values and the actual previous values in the sequence (when they're available). They are trained under the assumption that what comes next is influenced by what has come before. This makes them very powerful for doing sequential data predictions. In NLP, an auto-regressive model is a way of generating language [68].

The underlying statistical relationships in the training data are learned by these models. The models are thus enabled to produce not only coherent but also contextually fitting text. They adjust their output predictions as more words in a sequence are processed. This change keeps the language smooth and the tasks relevant for autoregressive models such as GPT, GPT-2, and GPT-3. These are autoregressive models that have established—and continue to establish—new benchmarks in natural

language processing [69]. GPT-3's unmatched ability to produce top-notch, human-like text and to carry out a dazzling variety of language-related tasks such as translation and SA placed it on a pedestal in terms of public perception [70]. While autoregressive models mainly predict future tokens using sequences from the past, they, too, have benefited from the advances in NLP that have come from the development of other models, such as BERT. What makes BERT interesting for our discussion is its ability to enhance the understanding of context in not just a single direction but in both of them, which makes it a much more powerful tool with which to penetrate the kind of context captured in tasks involving text dependencies [71].

2.3.1 Generative Pre-trained Transformer (GPT)

Yet another manifestation of NLP applications is chatbots. Chatbots are AI computers that carry on a human-like conversation using the text or voice processing methods of nearly half a century of research in human communication [72]. A chatbot, at least the consumer-facing kind, is a fairly basic piece of software. It is a user-interface layer over something that behaves like a brain with poor memory. When you ask a chatbot a question, the question goes either to a human sitting in some basement somewhere or to some supercomputer center's AI that has to work on the problem and find a solution [73, 74]. The ChatGPT language model has been trained and fine-tuned in a supervised manner and using reinforcement learning, to an extent not found with other chatbots. It is one of the largest and certainly the heftiest in terms of parameter count of the language models currently in use, with 175 billion parameters [75].

In terms of architecture, it is more than just a QA robot that can compose essays on a given subject in numerous languages. [76, 77]. The study in [78] used ChatGPT to summarize research based on the titles and other provided information of five high-reputation medical journals' research abstracts. ChatGPT worked well for creating scientific summaries from the gathered abstracts, and the plagiarism checkers might as well have been broken because they didn't catch anything. The study in [79] also employed ChatGPT in the business customer sector, specifically in the context of service analysis. It helped them pinpoint potential areas for improvement in the company's products and services, once again derived from an analysis of customer reviews.

2.3.2 Large Language Models (LLMs) for Financial Sentiment

The remarkable ability of LLMs to deal with domain-specific language has been demonstrated in the financial sector, where traditional models of SA often fall short. Given the unique hurdles posed by the almost impenetrable armor of technical language, abbreviations, and context-dependent phrases that are part and parcel of the

financial domain, one might assume that conventional models could handle the challenge posed by a more restricted problem space. Nonetheless, LLMs, having been fed massive amounts of data, have turned out to be the finest models for financial SA. LLMs perform exceptionally well when extracting sentiment from intricate financial texts, like earnings reports and market analysis, when they are trained on financial datasets. The LLMs are even better at picking up on the kinds of sentiment cues that are very easy for humans to miss, but traditional sentiment classifiers consistently misclassify if they classify them at all. For instance, when researchers [31, 80] at universities in Canada and the United States took a look at LLMs fine-tuned on financial training sets, they found that these models consistently outperformed earlier attempts at classifying sentiment in financial texts.

In terms of generalizability and adaptability to new and changing financial scenarios, LLMs hold considerable advantages. They learn extensively from pre-training data and identify patterns in financial sentiment that correlate with market movements or investor behavior. In prior research, LLMs like GPT-3 and FinBERT were shown to effectively process complex financial documents and predict stock price fluctuations through SA [81]. These models parse a significantly broader array of financial text sources than conventional ML methods. They are more adept at understanding the nuances of financial language, and therefore, they run sentiment-driven trading strategies using real-time market data with much greater success [82].

As researchers sought to understand the nuanced expressions and idiosyncratic language of finance-related writing, the area of financial SA began to take business. Yet even the best-fulfilling algorithm we have to currently, the pre-trained language model FinBERT, which massively outperforms our most universal models when we do any task related to finance, has proven unable to keep up with the language of the slowly but constantly evolving world of finance itself [81]. Meanwhile, a recent paper from Gite et al. [83] proposed a method of improving stock price prediction that fuses the stock market's sentiment, as expressed in the headlines of relevant news articles, with price-relevant data about the stocks themselves. Yet this approach, while it aims for only a narrow slice of the market, has shown to be highly reliant on historical data and offers an inadequate framework for the kind of real-time shifts in market sentiment that high-frequency traders obsess over.

Changing market conditions affect financial sentiment models, necessitating the development of models that can adapt to these changes without relying solely on static, pre-labeled datasets. Sharaff et al. [84] have taken the LSTM framework with word embeddings, which had already been converted into a financial news sentiment tool, and have pushed its performance into an even more favorable realm relative to many current models. This is, no doubt, due to the crucial role that context and temporal

dependencies play when attempting to extract meaning from the often dense and terse language of such articles. However, what appears at first blush to be a model with high performance is one with very limited applicability to the fast-paced world of finance due to its heavy reliance on long-term, semi-static historical datasets. Mishhev et al. [85] compared various methods of SA, especially the most advanced models. Even though today's transformer-based models perform really well, the authors pointed out that some genuinely fixable problems need to be addressed. They mainly involved the kind and amount of data available for training these models. But they also highlighted something else that I thought was important. And that's the issue of what you do with the model's output, particularly in real-time decisions that could have significant financial implications.

Additionally, present-day models often overlook the specialized financial vocabulary, which encompasses not only the distinctive terminology of the domain but also the abbreviations and shifts in sentiment driven by the context that make up the world of finance. Liu et al. [82] conducted a SA on social media data using a novel approach and, in the process, extracted investor sentiment from these otherwise undervalued reservoirs of information. After conducting a thorough validation of their tools, they found that the optimal way of interpreting sentiments from these networks is not as "positive", "neutral", or "negative", but instead in more contextually relevant categories that are "buy", "hold" or "sell". Ardekani et al. [86] introduced a financial sentiment model that uses contextual language processing, however, these scholars have yet to tackle the basic problems involved with aspect-based SA, which serves as the essential foundation for understanding the multi-layered kind of financial data we encounter.

2.3.3 Domain Adaptation Techniques

Techniques of domain adaptation are becoming ever more important for fine-tuning LLMs for specialized objectives, such as SA in the finance sector. TL is one of the most effective techniques in this context. It involves taking an LLM that has already been trained and using it as the basis for further training (referred to as fine-tuning) on a particular, much smaller set of text (the domain corpus). Right now, the most powerful LLMs available (e.g., GPT-3) have been trained first on an enormous amount of general text data. When we then take such a model and run it through a corpus of, say, specific financial filing statements, the model becomes much more capable in that context. That's still an oversimplification because it's not just the context that counts; it's also the model's comprehending of even the very simple words it uses to express the entity's sentiment.

The work of Shabbir et al. [56] demonstrates that TL has considerable advantages in performing SA in the finance domain. When they compared models that had

been first trained on a very large, general domain dataset (such as Wikipedia), and then fine-tuned on datasets from the finance domain, they found that these models outperformed those which had been trained only on general datasets from the start. When using this technique, LLMs are better able to capture the intricate and specialized language of finance, including sentiments expressed in earnings reports, vague forecasts, and general financial news. Doing so makes these models more accurately predict the kinds of things that models trained only on financial data have trouble with, and for good reason.

2.3.4 Evaluation Metrics and Benchmarks

Establishing the efficacy of SA models is vital for research and practice that allows one to make-side by side comparisons and undertake real-world model applications. Metrics used for model selection and assessment tend to be those that offer insight into the comprehensive efficiency of the model when it comes to correctly classifying sentiment. For our purposes, SA will be taken to mean the obligation of assessing the sentiment expressed in a provided content. Commonly used performance measures for SA models tend to be ACC, PR, RC, F1-measure, and AUC—ROC. More and more, however, SA model selection and assessment is based on nuanced, composite metric measurements that capture the abilities of SA models and their means of generating particular outputs for a given input [87]. An evaluation framework designed for context-sensitive LLMs that combine retrieval and generative methods such as RAGAS. RAGAS embodies the notion that for certain tasks like financial SA, both the search and the creating aspects of the model must work together seamlessly for the model to perform well. RAGAS includes metrics tailored to the dual context/retrieval performance aspect of the model with selected components that assess the answer relevance, answer correctness, answer similarity, answer precision, and answer recall of the provided model [88].

Moreover, PTLMs for SA tasks are evaluated based on AAC, F-Measure, RC, and PR. While these metrics are generally accepted within the broader LLMs and machine-learning communities, they often fail to capture the intensity and granularity of a statement in human understanding, particularly in financial text (e.g., online social media text) in for the purpose of identifying whether the sentiment is pos, neg, or neu [89, 90]. To address this gap, our study uses new metrics called RAGAS combined with sentiment intensity, the level of detail in sentiment classification, and the Vendi Score with ARLMs such as Llama-2¹⁷, Llama-3¹⁸, OpenAI GPT-3.5-Turbo¹⁹ and OpenAI GPT-4o-Mini²⁰ that have the capability to estimate the next pattern of

¹⁷<https://llama.meta.com/llama2/>

¹⁸<https://llama.meta.com/>

¹⁹<https://platform.openai.com/docs/models/gpt-3-5-turbo>

²⁰<https://platform.openai.com/docs/models/gpt-4o-mini>

expressions in a sentence or text, and they have significantly expedited recent developments in NLP tasks like SA.

2.4 Hybrid Models for Real-Time Sentiment Analysis

We look at hybrid models in this section. Hybrid models are models that combine different advanced techniques to yield improved performance, especially in real-time applications of SA introduced by Zhang et al. [91]. They integrated DL with traditional ML to yield better results, in this case, for sentiment classification of financial texts, judging by some nearly perfect results they reported with an ACC of 94.79%. This model took advantage of both the features of the domain and the general patterns of language, which made their predictions more reliable under market conditions. In much the same way, Lewis et al. [92] implemented a hybrid system using RAG, which incorporated external knowledge bases into the generative process. As such, they tackle a basic problem with using a LLM in a volatile setting.

A key evolution in hybrid SA models is the use of TL. The potential that TL has for these models was demonstrated by Rehan et al. [93], who fine-tuned pre-trained models on a minimal amount of domain-specific data, achieving cutting-edge results in the sentiment classification of specialized domains like healthcare and customer service, where labeled training data is often limited. These models are now being used in real-time for SA in a number of sectors.

2.4.1 Fine-Tuning

Adapting LLMs specifically for SA tasks allows for more reliable and significant results. The architecture of these models is so vast that when they are fine-tuned, it is usually a pre-trained model that is fine-tuned on a limited amount of specialized data that is highly relevant to the domain of interest such as finance, healthcare, hospitality, etc. This amounts to a fine-tuning not just of the model's parameters but also of the sort of understanding the model is supposed to have in the domain in which it is being applied. An instance of this can be found in financial SA, in which models are fine-tuned on the kinds of language found in financial reports, articles, and social media. From such data, they learn to predict the kinds of shifts in sentiment that accurately anticipate market movement. Du et al. [94] showed that using domain-specific materials to fine-tune pre-trained models like GPT, with the model's earlier layers frozen as they were in the past, considerably boosts performance over using the model in a more general way.

The field of healthcare SA has benefited from the technique of fine-tuning, where pre-trained models such as BERT and RoBERTa have been applied to understand the precise and often complex language of the medical domain. Model fine-tuning

on healthcare datasets has led to critical enhancements in understanding the subtleties of patient sentiment expressed in health records and other forms of electronic communication with their providers. For instance, Huang et al. [95] demonstrated the utilization of the fine-tuned BERT model for this very task, achieving, to my knowledge, the best results reported to date for electronic health record SA. This approach is almost certainly generalizable to other similar tasks; witness the modestly enthusiastic effectiveness claim in the field.

2.4.2 Retrieval-Augmented Generation (RAG)

Substantial development has occurred in SA based on RAG models that combine the advantages of retrieval-based models with those of generation-based models. A very simplified explanation of a RAG model would lead one to think that a two-step process occurs. The first step is to retrieve relevant contextual information from an external knowledge source, like a database that contains accurate information about the present and the near future. In the second step, the model uses the RAG stored context and augments it via a generating system to produce sentences that gathering the relevant sentiment of the context. Although the study by Lewis et al. [96] does delve deeply into the mechanism by which context is fed into the RAG model, one could assume there is some suitably crafted prompt that evokes a right-here-right-now way of thinking.

In addition, RAG models effortlessly integrate external data into text generation. That quality makes them ideal for time-sensitive applications, like news SA or social media monitoring. RAG's ability to produce up-to-date sentiment assessments practically guarantees the ACC and reliability of sentiment forecasts that allow developments to continue to unfold. Karpukhin et al. [97] demonstrated that retrieval-augmented methods could significantly boost the performance of SA models when they're called on to operate in real-time. This makes RAG a critical tool for industries that rely on up-to-the-minute SA, such as finance and politics.

2.4.3 Transfer-Learning (TL)

Using minimally labeled data, we have found that TL can match even direct optimization on an analysis task, like sentiment prediction, in terms of model ACC. This approach involves taking a model that has been trained on a broad, general dataset and then refining it with a somewhat narrow and less plentiful domain where labeled data is available but costly to acquire—like the finance sector. We leverage the general knowledge the model has learned and adapt it to new, domain-specific language and tasks. In the domain of financial SA, Vulic et al. [98] highlight the effectiveness of this method, achieving greater ACC, not just paywall in the Moore case, than a direct optimization of our task at hand in the secure domain.

Furthermore, healthcare SA has greatly benefited from the application of TL, a method that compensates for the scarcity of well-annotated medical data owing to privacy issues and the expense of producing such data. To illustrate, Lee et al. [99] fine-tuned BERT models for clinical sentiment tasks and found that their performance was much improved relative to their predecessors. The study demonstrated that the BERT model could be fine-tuned to achieve cutting-edge effectiveness on certain clinical SA tasks, like understanding and assessing patient feedback left in the structure of free text.

A detailed comparison of the studies presented so far is provided by Table 2.2. The table summarizes the studies along several dimensions: model type, feature extraction methods, performance metrics, and study gap. From this comparison, we can see not only the progression of techniques but also the growing importance of real-time, context-sensitive, and domain-specific SA models, fine-tuned for the kinds of applications we might encounter in areas like finance.

2.5 Summary

A thorough summary of the SA literature has been stipulated in this section. Both traditional and contemporary techniques used in the field have been covered. We went back to explore the antecedents of today's cutting-edge methods, starting with dictionary-based approaches. These are understandably the go-to tools for many in the social sciences, given their simplicity, interpretability, and generally decent performance across a number of experimental contexts. However, they often leave something to be desired in terms of handling language that is informal, context-dependent, and rife with domain-specific terminology, especially in such dynamic and complex applications as "analyzing" sentiment on social media platforms.

We also looked at how SA techniques have evolved with the developments of DL and LLMs. It recounted how these foundational layers of our AI landscape, now technologies in their own right, have been used to create more effective means of computing sentiment from text. Core to this revolution in our ability to analyze sentiment from text have been the various word embedding models and the recurrent neural nets used to process language and to layer it in ways that capture context and the all-important semantic relationship among words. RNNs have been bested by newer and fancier models, to the point that industry seems to have almost forgotten about RNNs. Yet, what matters most for business applications is these models' superior ability to compute the context in which words are used. Having established

TABLE 2.2: Analyzing studies with shared characteristics for comparison

Reference	Features	Model	Metric	Dataset	GAP
[52]	TBoC	NB, SVM and NN	ACC, FM, PC and RC	SemEval	Lack of DL methods
[53]	Frequency, Unigram	linear SVM	ACC & Cross-validation	Twitter (now X)	Limited feature diversity
[60]	Frequency count, n-gram	BoW	Ratio	Hotel	Small dataset, lacks advanced models
[100]	n-gram, BoW and PSO	SVM and ADASYN	ACC, AUC and FM	Restaurant Reviews	No exploration of DL
[57]	TF-IDF	RF-C, SVM-C, GB, XGB	ACC, and FM	Twitter (now X)	Limited TL applications
[101]	TF-IDF	DIC, NB, SVM, ME, and CRF	ACC	Twitter (now X) and Amazon	Lack of neural network models
[102]	Skip-gram, CBOW	SVM, RF, LR and SGD	F-Measure	LABR, ASTD, MPQA	No comparison with transformer models
[54]	n-gram model	Voting methods	ACC	Amazon website	Limited ensemble techniques explored
[103]	skip-gram	RF	ACC, PR, RC and F-Measure	Twitter (now X)	No evaluation on diverse datasets
[104]	LoRA Fine-tuning	Llama-2	ACC, F-Measure	Custom datasets	Limited generalization across datasets
[105]	LLMs, TL	GPT-3, BERT	ACC, PR, RC	Financial datasets	Lacks real-time evaluation
[106]	LLMs, Transformer	GPT-4, ChatGPT	ACC, F-Measure	Open-source datasets	Insufficient dataset diversity
[79]	ChatGPT, SA	ChatGPT	ACC, F-Measure	Customer reviews	No cross-domain performance
[107]	LLMs, Multimodal Learning	GPT-4, Multimodal Models	ACC, F-Measure	Multimodal datasets	Limited exploration of multimodal data fusion
[108]	LLMs, Text Generation	GPT-3, GPT-4	BLEU, ROUGE	Text corpora	Lack of human evaluation metrics
[109]	Retrieval-Augmented LLMs	GPT-4, LLaMA	ACC, F-Measure	Financial news datasets	Insufficient generalization to other domains
[106]	LLMs, Ethics	GPT-3, GPT-4	Qualitative Analysis	N/A	No quantitative evaluation methods
[6]	LLMs, Reinforcement Learning	GPT-3, GPT-4	ACC, F-Measure	Various NLP benchmarks	No exploration of real-time learning

the theoretical foundations and examined various models' contextual computing capabilities, the next chapter delves into data acquisition and analysis to apply these insights practically.

Chapter 3

Dataset and Exploratory Analysis

The chapter describes how the datasets used for financial SA were collected and prepared. The main source of data was customer reviews from the Hellopeter website, focusing on comments about the major financial institutions listed on the JSE. These reviews were used to build a primary dataset. To provide the sentiment expressed in those reviews with a richer context, we also sought out and used cross-domain datasets. The exploratory analysis presented in this chapter reveals some interesting aspects of the datasets and provides a basis for understanding the kinds of things that the datasets can lead us to interpret.

3.1 Dataset

This study uses a dataset that consists of three primary parts: customer review data from Hellopeter, data from the top five financial institutions on the JSE for five years, and cross-domain datasets from various sources. The datasets together serve as a big dataset for conducting a comprehensive SA of the financial sector. More importantly, the dataset enables us to perform not just an analysis of the customers' sentiment but also one of the public's overall sentiments toward the financial sector since it contains a component that can be termed real-time customer experience sentiment. This aspect of the dataset allows us to key in on what's happening in real-time in terms of the virtual engagement in sentiment by the public toward the financial sector.

3.1.1 Customer Reviews Dataset

The dataset of consumer reviews was obtained from Hellopeter, a prime platform for customer interaction based in South Africa. When consumers wish to convey an experience in dealing with a business, they go to this platform and leave messages that are then seen by the world at large. Hellopeter customers post these types of reviews because they usually want something to happen as a result, either a response from the business they are writing about or some sort of internal action taken by the business that leads them to the review platform in the first place. For this study, the dataset focuses on reviews related to five top South African financial institutions:

TABLE 3.1: Summary of customer reviews dataset and tokenized features for each financial institution.

Dataset	Size	Positive	Negative	Token	Features
ABG	20,286	8,466	11,820	2,780,995	35,942
SBK	22,776	9,530	13,244	3,022,224	35,359
CPI	16,456	6,485	9,971	2,084,806	28,960
FSR	54,900	24,338	30,561	6,865,630	54,547
NED	14,923	6,026	8,897	2,013,942	29,343

ABG, SBK, CPI, NED, and FSR. The data were collected over a five-year period, from January 2018 to December 2022, using Hellopeter’s publicly available API¹.

The pulled data features encompass the author’s details such as ID, review date, rating, content, and business name, among other relevant particulars. Table 3.1 gives an overview of the dataset collected for each of the financial institutions, summarizing the number of tokens in the review business documents and the features extracted from those documents.

In order to better the dataset, we classified each review by its financial institution: ABG, Capitec, SBK, NED, and FSR. We then applied OpenAI’s ChatGPT to the zero-shot learning model for annotation of review sentiment. The advantage of using ChatGPT is that it employs cutting-edge NLP capabilities to determine whether the sentiment expressed in each sentence was positive or negative. ChatGPT’s decision process for classifying each sentence was guided by a pre-trained language model, which assigns learned patterns in the text to assign sentiment labels.

After the process of annotation was complete, the reviews were preserved in separate files, each fillet corresponding to a specific financial institution. For each institution, we generated a single CSV file. Each of these files contains two columns. One column holds the text of the review, while the other holds a label indicating the review’s sentiment class that has been assigned to it (positive or negative). Table 3.1 shows how many reviews of each sentiment class we obtained for each of the financial institutions after we did some preprocessing.

To enhance the feature selection process, we pruned the dataset by removing stop-words, duplicates, and non-English characters. This cleaning step was essential to owning an assembly of features that were not only relevant but also necessary for establishing a significant and coherent SA. Following this, we employed the BOW model for feature extraction. Unlike the previous method of no feature extraction, which yielded perfect classification ACC for the training set, the BOW model seemed much more conducive to yielding a generalizable and significant result for SA in the Hellopeter reviews.

¹<https://business.hellopeter.com/docs/api/v5>

3.1.2 Financial Data

Inherently volatile, the stock market makes forecasting future stock prices a daunting task. For reliable financial data to enlighten our analysis, we enlisted the Yahoo Finance APIs² [110] and the Finchat APIs³ [111], both of which are adeptly integrated with the Stratosphere platform. This combination of tools allowed us to unearth unique identifiers for the top banking companies listed on the JSE⁴. To diversify our dataset, we gathered macroeconomic indicators, such as the Consumer Price Index and the Financial Stability Review. Our data collection commenced in earnest in 2018 and progressed through 2022. However, our dataset was augmented in 2023 for certain institutions, such as FirstRand and CPI, as shown in Table 3.2 that had already released their 2023 figures and thus allowed us to maintain the topicality of our analysis.

We ensured that the financial data we collected were reliable and accurate by cross-verifying them with official bank financial statements [112–116]. This process helped to ensure that our array of stock market data was consistent with the data the banks published in their financial reports. The following analysis uses data from the five major South African banks: ABG, SBK, CPI, NED, and FSR. We were able to get our hands on some critical figures that needed understanding of the banks' market sentiment and stock price fluctuations of daily and monthly stock prices. Figure 3.1 and 3.2 shows the price distributions across the five banks, for both daily and monthly appearances, across our observation period.

3.1.3 Selection of Financial Indicators

We chose our study's financial indicators with care, aligning them closely with our research objectives and ensuring we selected key metrics that convey the most pertinent, reliable, and insightful evidence of the financial performance and overall health of the sampled institutions. Our indicators concentrate almost entirely on highly meaningful profitability metrics. At the same time, they include some essential components of our latest risk management and financial stability indicators, such as the loan loss provision, which is a heck of a thing for a banking institution to have to write down, which indicates you aren't doing well, and non-interest income, which is high in the indicators of the greater banks we've seen in recent years and thus means, to a certain degree, that those institutions aren't appearing to use their core profit-making function interest income efficiently enough [117].

Besides these financial indicators, stock market data enrich our analysis, specifically daily, monthly, and yearly price trends, as shown in Figure 3.1 and 3.2. This stock price data lets us look at the interplay between market dynamics and financial performance. We incorporate both internal financial metrics and external market data

²<https://developer.yahoo.com/api/>

³<https://finchat.io/api/docs/>

⁴<https://www.jse.co.za/>

TABLE 3.2: Summary of income statement metrics for each financial institution (in millions).

Symbol	Metric (in Millions)	2018	2019	2020	2021	2022	2023
ABG	Interest Income, Total	89,236.00	97,838.00	93,051.00	89,495.00	112,232.00	null
ABG	Interest Expense, Total	45,481.00	51,337.00	44,194.00	36,173.00	51,734.00	null
ABG	Net Interest Income	43,755.00	46,501.00	48,857.00	53,322.00	60,498.00	null
ABG	Non-Interest Income, Total	32,722.00	33,607.00	32,741.00	32,519.00	38,315.00	null
ABG	Total Revenues	70,153.00	72,292.00	61,029.00	77,342.00	85,110.00	null
SBK	Interest Income, Total	129,582.00	125,863.00	108,673.00	100,753.00	135,626.00	null
SBK	Interest Expense, Total	68,561.00	61,024.00	45,600.00	36,776.00	56,484.00	null
SBK	Net Interest Income	61,021.00	64,839.00	63,073.00	63,977.00	79,142.00	null
SBK	Non-Interest Income, Total	65,596.00	66,396.00	68,340.00	68,769.00	77,900.00	null
SBK	Total Revenues	120,128.00	123,271.00	110,819.00	122,873.00	144,978.00	null
NED	Interest Income, Total	75,941.00	83,680.00	72,300.00	65,772.00	82,104.00	null
NED	Interest Expense, Total	47,122.00	53,513.00	42,219.00	33,272.00	45,827.00	null
NED	Net Interest Income	28,819.00	30,167.00	30,081.00	32,500.00	36,277.00	null
NED	Non-Interest Income, Total	25,947.00	25,979.00	23,225.00	25,645.00	28,516.00	null
NED	Total Revenues	51,078.00	50,017.00	40,179.00	51,611.00	57,412.00	null
CPI	Interest Income, Total	null	15,501.10	17,041.30	16,544.30	17,454.30	21,199.30
CPI	Interest Expense, Total	null	4,509.50	5,679.80	4,984.40	4,837.90	6,992.70
CPI	Net Interest Income	null	10,991.50	11,361.40	11,559.80	12,616.40	14,206.60
CPI	Non-Interest Income, Total	null	8,017.80	9,723.00	11,311.80	14,257.80	16,093.80
CPI	Total Revenues	null	14,559.10	16,610.70	15,047.00	23,366.50	23,971.00
FSR	Interest Income, Total	null	120,268.00	121,887.00	105,935.00	108,967.00	151,896.00
FSR	Interest Expense, Total	null	64,562.00	63,757.00	46,971.00	48,547.00	84,790.00
FSR	Net Interest Income	null	55,706.00	58,130.00	58,964.00	60,420.00	67,106.00
FSR	Non-Interest Income, Total	null	50,559.00	46,450.00	49,609.00	54,222.00	64,670.00
FSR	Total Revenues	null	95,765.00	80,197.00	94,913.00	107,562.00	120,827.00

to provide a more consistent evaluation of the institutions’ combined financial resilience, market behavior, and growth prospects [118, 119]. The basic tenet here is that if an institution is truly healthy and has worthwhile growth prospects, it should be reflected both in the internal metrics we can measure and in the external market data we can observe.

3.2 Data Acquisition and Processing

This section describes in detail the process used to acquire and prepare the datasets for this study. We concentrate here on the methods used to obtain the data. We also describe the relatively few steps we took to preprocess the data, which helped with both the quality and consistency of the datasets. That’s two classes of problems that we head off in order to ensure robust SA.

3.2.1 Data Acquisition from HelloPeter

The main dataset was gathered from HelloPeter, a top consumer review platform in South Africa that lets users write about their experiences with financial services. We utilized API endpoints to extract customer reviews pertaining to five of South Africa’s largest banks—namely SBK, NED, ABG, Capitec, and FSR—that we fetched with the help of a Python script. In addition to these bank-specific reviews, we incorporated some cross-domain datasets that allowed us to train the model under

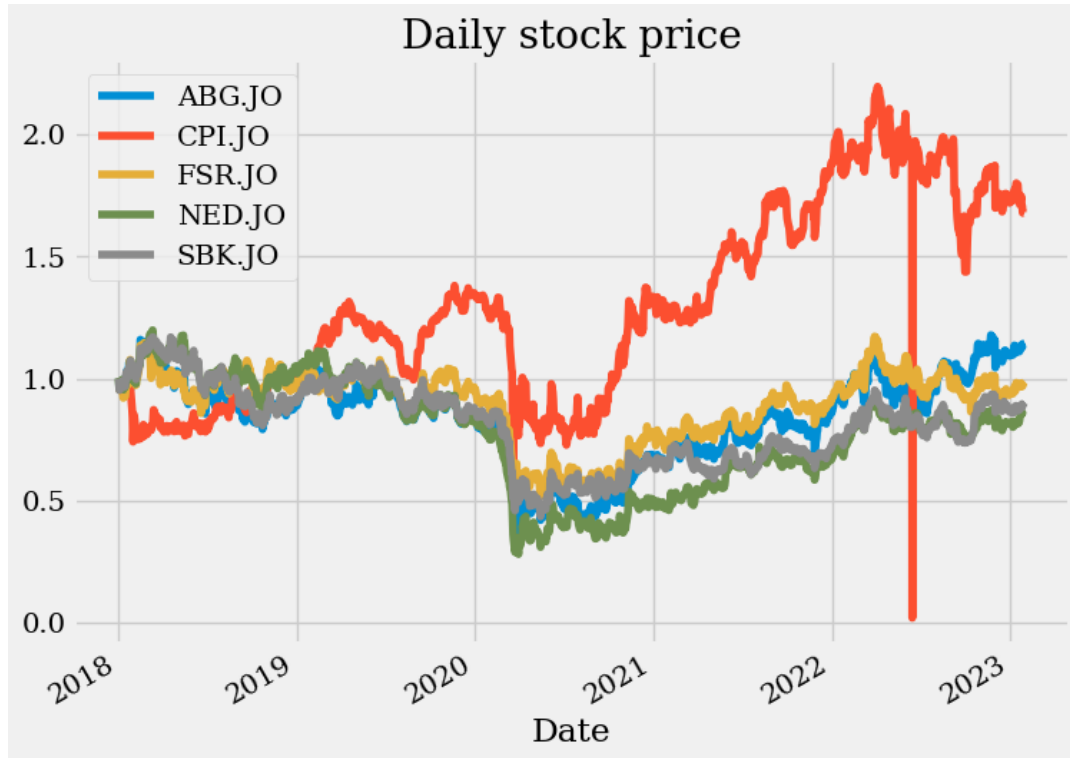


FIGURE 3.1: Stock price distribution of daily stock prices for top South African banks.

very diverse contexts and with very diverse sentiment expressions, potentially increasing the chances that it will outperform the baseline when it comes to analyzing bank-specific reviews in this or that context. The two datasets, one bank-specific and the other cross-domain, are summarized in Table 3.2.

We applied an efficient data extraction algorithm to ensure adaptability and relevance across the different domains of our model. This algorithm 1, which runs as a series of steps, uses multiple API endpoints to get review data from Hellopeter. Once retrieved, the data is cleaned, when necessary, for noise and errors; it is also formatted and filtered so that only the most relevant parts are retained. The relevant parts include the review date, a title (when there is one), and the actual content of the review. This content is then used to apply SA.

The Hellopeter dataset is summarized in Table 3.3, illustrating how the data are divided among the training, validation, and test sets. In total, 129,341 reviews were collected when combined, having a token count of over 8 million. Thus, the data are as diverse as they are plentiful, making for an ample and somewhat more polished dataset for performing SA.

TABLE 3.3: Hellopeter Dataset Summary

Dataset	Train	Validation	Test	Total Size	Tokens
Hellopeter Reviews	103,472	12,934	12,935	129,341	8,340,421

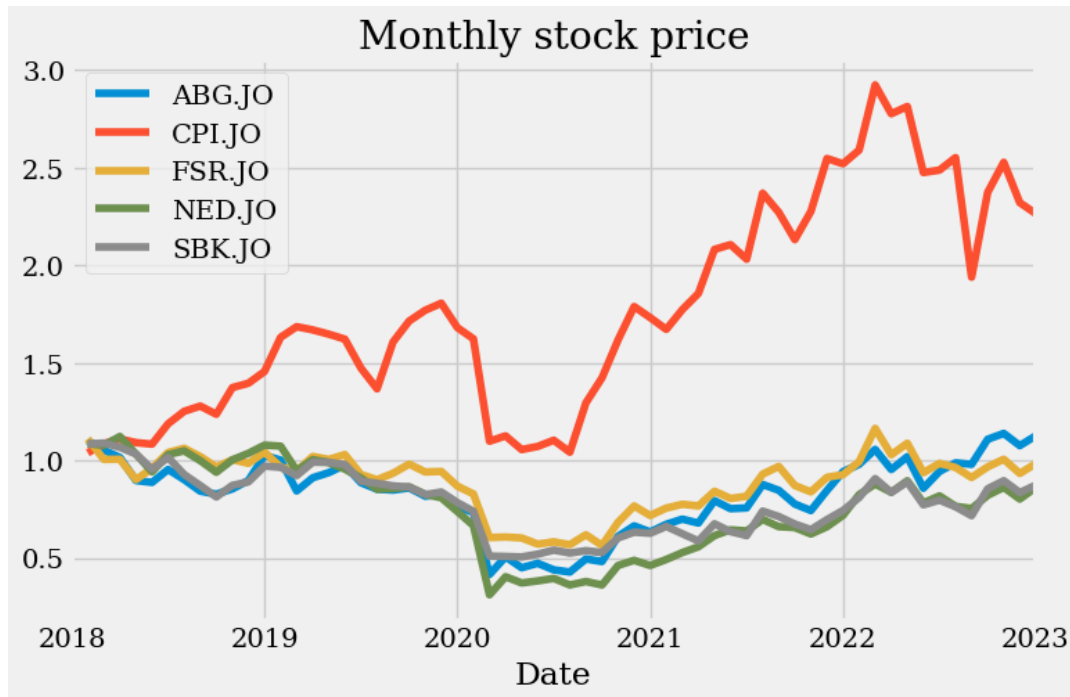


FIGURE 3.2: Stock price distribution of monthly stock prices for top South African banks.

The regained information was then subjected to another layer of preprocessing that ensured quality and ready-to-use data for the kind of analyses performed in this work. Procedures employed in this layer included, first and foremost, the removal of duplicates and irrelevant information. Non-English characters such as symbols, emojis, and stopwords were also removed. After this, a series of text normalization techniques was applied that helped improve the consistency of the dataset. These steps were taken so that the features extracted from the data would be the most usable, meaningful, and important ones, leading to a correct estimation of the performance of the SA tasks.

3.2.2 Cross-Domain Datasets

This study's main data were gathered from the Hellopeter website, which is one of the most popular platforms in South Africa for documenting customer experiences with service providers, including those in the financial sector. Experiences with major South African banks like SBK, ABG, and FSR were accessed through API endpoints. The nature of the customer experiences provides a clear window through which to view the performance of these banks at the level of customer sentiment.

Alongside the Hellopeter data, the analysis incorporated cross-domain datasets into the model to improve its generalizability and to furnish a context for broader sentiment expression. These datasets were drawn from several domains, such as restaurants, movies, hotels, politics, and finance, providing a richer, more varied sentiment landscape. The system architecture the study employs is highly extensible,

Algorithm 1 Fetching and Processing HelloPetter Reviews

Input: List of API endpoints (URLs), Maximum pages to retrieve ($P_{\max}$)

Output: Collection of processed reviews $R_{\text{processed}}$

```

1: Initialize  $R_{\text{processed}} \leftarrow \{\}$ 
2: for each  $url$  in URLs do
3:    $response \leftarrow$  Fetch initial data from  $url$ 
4:    $P_{\text{total}} \leftarrow \min(\text{Extract total pages}, P_{\max})$ 
5:   for  $p = 1$  to  $P_{\text{total}}$  do
6:      $data \leftarrow$  Fetch data from page  $p$ 
7:     if data is valid then
8:       Extract relevant fields:  $created\_at$ ,  $review\_title$ ,  $review\_content$ , etc.
9:        $review \leftarrow$  Combine relevant fields into a single text string
10:       $R_{\text{processed}} \leftarrow R_{\text{processed}} \cup \{review\}$ 
11: return  $R_{\text{processed}}$ 

```

TABLE 3.4: Cross-domain datasets used in model training and evaluation.

Dataset	Domain	Total Size	Tokens
SemEval-2014[120]	Restaurants	2,503	54,086
IMDB-50k [121]	Movies	50,000	13,974,161
TripAdvisor [122]	Hotels	20,491	2,389,961
SemEval-2016 [123]	Politics	26,750	604,358
Financial PhraseBank [124]	Financial News	2,264	50,891

which allows for the easy integration of multiple data sources through application programming interfaces (APIs) or file formats. This architecture’s scalability means the model can shift across different domains and still serve a useful purpose.

Incorporating cross-domain datasets delivers a powerful boost to the training and assessment of SA models, which gauge the positive or negative of user sentiments in the diverse contexts of product and service reviews. The included data bolster the models with new instances of various sentiments expressed in many different ways. For example, the SemEval-2014 dataset provides restaurant reviews; IMDB supplies movie reviews; and SemEval-2016 gives us political stance content for a more comprehensive understanding, which is the SA: A Cross-Domain Perspective model. Table 3.4 provides a summary of the cross-domain datasets used, detailing their respective domains, sizes, and token counts.

3.2.3 Data Preprocessing and Feature Extraction

Before we conducted SA, we undertook an extensive data preprocessing stage on all datasets. This was necessary to achieve consistency, reduce noise, and get rid of irrelevant features. To carry out this, we implemented several key steps: We removed stop words, eliminated non-English characters, and normalized the text. We also tokenized the input so that we could work with the individual terms or tokens

more effectively. Overall, we felt that this was a vital part of the process and a good foundation for the feature selection and extraction stage that followed.

We employed two fundamental methods for feature extraction: the BoW model and N-grams. The BoW [125] method treats each text document as a collection of unique words while maintaining word frequency. It conveys the idea, that if we only had a few more unique words in this collection, we could get a better representation of the underlying sentiment. N-grams [126] are word sequences that can capture a little more contextual information and therefore work better. We also applied the TF-IDF [127] method to assign quantities to the all-important terms within the dataset that best express the underlying sentiment.

After feature extraction and preprocessing, the dataset was partitioned into training, validation, and test subsets. They positively and negatively balanced all partitions to ensure that across all three subsets, there was a clear expression of sentiment. This was done because our aim was not only to have the model perform well on the reviews it was trained on but also to have good generalization across all types of reviews.

3.3 Visual and Statistical Interpretation

This section describes the various statistical and visualization methods used to analyze sentiment distribution across customer reviews. An initial examination of these visual representations provides valuable insights into the underlying patterns within the data. Certain trends and relationships emerge, some aligning with expectations while others may be unexpected. Additionally, these visualizations help identify correlations between sentiment patterns and key financial performance indicators. By employing these methods, we gain a deeper understanding of whether a meaningful relationship exists between sentiment appearance and institutional performance.

3.3.1 Univariate Distributions

To probe the single-variable distributions of sentiment in customer reviews, we use a technique called probability binning (PB) [128]. This divides the distribution into a set number of bins. Whereas traditional binning algorithms use bins of equal width, PB adjusts the binning such that each bin contains an equal number of events. So, PB doesn't treat the "bins" as bins at all but treats them as bins of equal probability. The advantages of this method are twofold. First, it allows for more meaningful comparisons across bins when we perform statistical tests like chi-squared calculations, and second, it doesn't force our narrative into an equally divided space. That space is, in fact, more narratively impressionistic than any amount of flag waving faithful to the original reviews could convey.

Customer review sentiment polarity is counted in the 0.05-interval bins of sentence length. The plot in Figure 3.3 shows how the review sentiments are spread throughout the dataset. As shown, some reviews are quite long in comparison to others, and many reviews, regardless of length, are quite extreme, either positively or negatively.

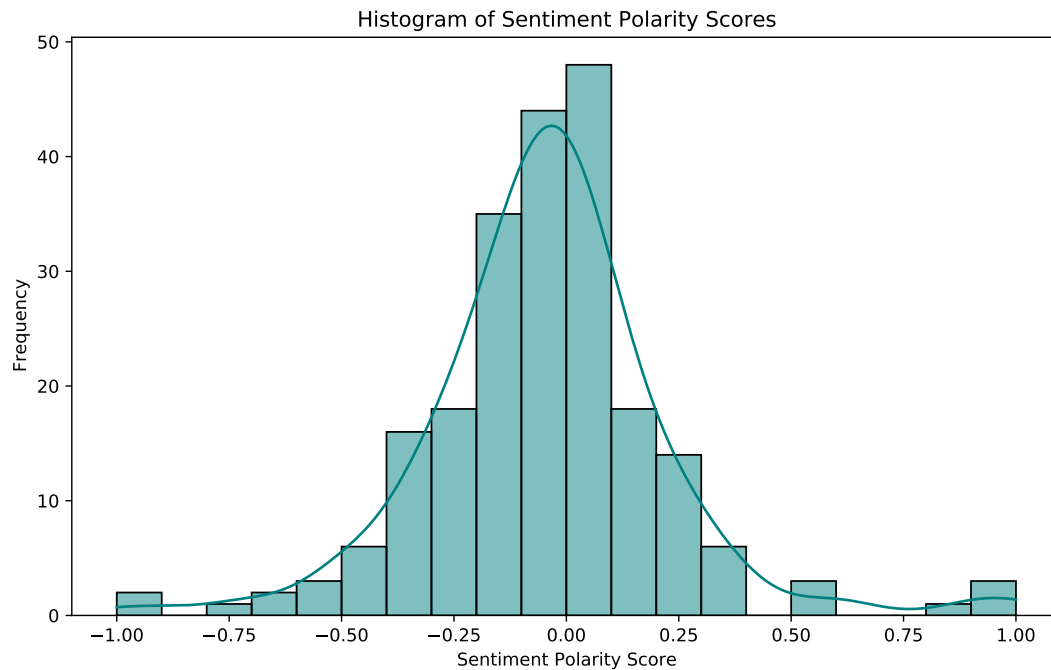


FIGURE 3.3: Distribution of sentiment polarity counts across sentence lengths.

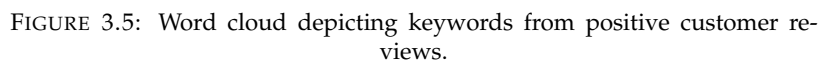
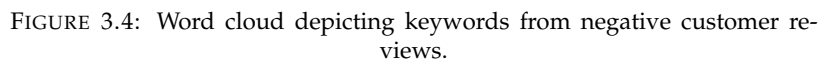
Each training dataset undergoes PB using 10, 20, 30, or 40 bins, and the mean and standard deviation of $1,000 \chi^2$ values are calculated for sentiment polarity scores. The resulting χ^2 distributions exhibit near-normal behavior, indicating that the standard deviation can be used to measure the variance of the χ^2 values.

In the dataset from Hellopeter, positive customer reviews mostly center around mid-range polarity scores. The average for the positive reviews is around +0.5. The negative review scores are mostly banded around their negative average of about -0.5. When you look at the distribution of the scores, it's pretty narrow, and the reviews tend to express not only a balanced sentiment but also a sentiment that's moderate rather than extreme.

3.3.2 Word Cloud Visualization

To help interpret the most frequently occurring underlying sentiments in customer reviews, we used word clouds. By visually displaying the prominence of words based on their frequency, a word cloud gives you the impression you need to get the key theme surrounding the dataset you're looking at to allow quick understanding of your key elements in the dataset. In our analysis for this report, a maximum of 300 words appear in each cloud. Any individual word that appears in the customer

Figure 3.4 and 3.5, illustrates the most common words used by consumers when expressing positive and negative sentiments about financial services. These visualizations allow us to quickly identify the key areas that customers either appreciate or find unsatisfactory, offering valuable insights into potential improvement areas. The positive and negative term clouds highlight different aspects of financial services that are deemed either good or bad, occurring frequently enough to catch the average viewer's attention.



3.3.3 Scattertext for Phrase Visualization

We use Scattertext [129] to analyze customer review data from our applications. This tool enables us to identify trends and extract meaningful insights from the reviews. One of the primary reasons for selecting Scattertext is its effectiveness in visualizing phrases that are strongly associated with or distinctive to specific reviews. These visual representations help uncover patterns in customer feedback, providing a clearer understanding of the key linguistic elements that shape sentiment analysis.

Scattertext constructs a set of unigrams and bigrams (referred to as “terms”) found in a corpus C with disjoint subsets A and B such that $A \cup B \equiv C$. The probability of a term t_i occurring in the corpus is given by equation 3.1:

$$P_r[t_i] = \frac{\phi(t_i, C)}{\sum_{t \in C, |t| \equiv |t_i|} \phi(t, C)} \quad (3.1)$$

where: - $\phi(t, C)$ is the frequency of term t in corpus C , - $\sum_{t \in C, |t| \equiv |t_i|} \phi(t, C)$ normalizes based on terms of the same length.

To filter out bigrams that do not occur significantly more frequently than expected, Scattertext employs Pointwise Mutual Information (PMI) as equation 3.2:

$$PMI(t_i) = \log \frac{P_r[t_i]}{\prod_{t_{ij} \in t_i} P_r[t_{ij}]} \quad (3.2)$$

where: - t_i is a bigram, - t_{ij} represents individual words in the bigram t_i , - $P_r[t_{ij}]$ is the probability of individual words in the corpus.

The final set of terms included in the visualization V is defined as equation 3.3:

$$V = \{t \mid \phi(t, C) \geq m \wedge (|t| \equiv 1 \vee PMI(t) > p)\} \quad (3.3)$$

where: - m is the minimum frequency for a term to be included, - p is the minimum PMI threshold for bigrams.

The coordinates of a term t in the scatterplot for categories A and B are given by equation 3.4

$$x_K^t = \frac{r_K^t}{\max r_K} \quad (3.4)$$

where: - r_K^t is the rank of term t in category K ($K \in \{A, B\}$), sorted in ascending order, - $\max r_K$ is the highest rank, ensuring x_K^t values remain in $[0, 1]$.

Finally, the “scatter score” that determines how strongly a term is associated with a category is given by equation 3.5:

$$s_K(t) = \begin{cases} \|(1 - x_A^t, x_B^t)\|, & \text{if } K = A \\ \|(x_A^t, 1 - x_B^t)\|, & \text{if } K = B \end{cases} \quad (3.5)$$

where terms closer to the lower-right corner are more frequent in A than in B , and vice versa.

These equations form the basis of Scattertext's term visualization approach, allowing users to explore key distinguishing phrases in a dataset. We implemented a model using Python's "numpy", "pandas", and "collections" libraries to analyze the frequency of words and phrases, enabling the detection of shifts in subject matter using Scattertext and a scaled F-score. Figure 3.6 displays the distribution of phrases in customer reviews from Hellopeter. Figure 3.6 presents noun phrases generated using the BoW model.

The traits linked to a positive sentiment are located higher on the y-axis, nearer the upper left of the plot, while the features expressing negative sentiment are positioned further to the right on the x-axis, near the lower right corner. The axes might as well be labeled positive sentiment and negative sentiment. The most frequently used phrases of either sentiment fall somewhere in the upper half of the plot. Phrases that tend to express a more intense positive or negative sentiment are located closer to the upper left and lower right corners of the plot.

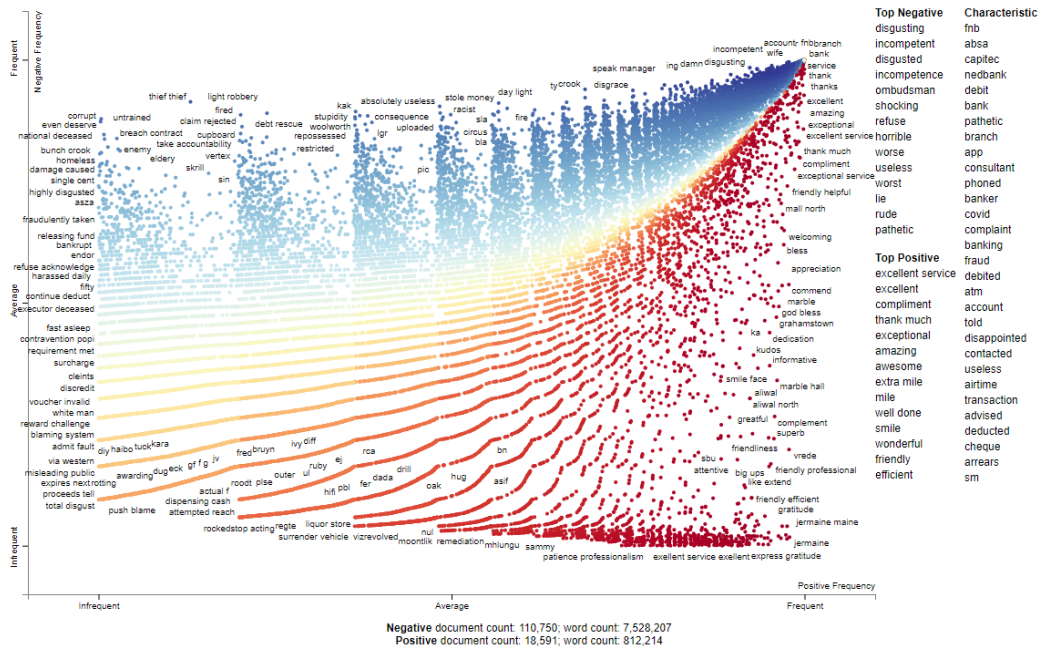


FIGURE 3.6: Visualization of sentiment categories using Scattertext. Red indicates negative sentiment feedback, while blue represents positive sentiment, with color intensity reflecting frequency.

As shown in Figure 3.6, the most frequent terms associated with negative sentiment in Hellopeter reviews include phrases such as "disgusting", "shocking", "lying",

“rude”, and “pathetic”. Conversely, positive sentiment reviews frequently feature terms like “excellent”, “awesome”, “amazing”, “friendly”, “efficient”, and “compliment”. Some terms, such as “exceptional service”, “kudos”, and “salute”, appeared frequently across both sentiment categories, suggesting broad consumer satisfaction or dissatisfaction with the same services. Further shows that customer frustration often colored the negative sentiment phrases. Terms like “threat”, “market link”, and “priority pass” were some of the words customers used to express their discontent. On the other hand, when positive phrases were used, they mostly contained expressions of happiness such as “great experience” and “excellent service”. The analysis makes it pretty clear where customers did not have their expectations met and where they did, making it a useful tool for assessing service quality.

3.4 Summary

This chapter collected, preprocessed, and analyzed a set of customer reviews and a set of financial data. To improve the cross-domain generalizability of the sentiment model we built, we integrated two basic, but disparate, types of data: what customers said and what financial metrics said. Structuring and understanding those two types of data involved some fairly standard visual and statistical analysis. We offered both kinds of analysis as a way to get inside what is a fairly complex and not very intuitive relationship between the two domains of data. Building on this foundational understanding of data analysis, the next chapter will explore how these insights can be leveraged to enhance text annotation for sentiment analysis, focusing on the use of ChatGPT as a tool in this domain.

Chapter 4

ChatGPT as a Text Annotation Tool for SA

Today, social media offer a real-time, participatory way for people to express their thoughts and feelings about financial products and services—an almost instant way to paint a picture of customer sentiment. For any person considering a major financial decision, the immediacy and depth of sharing on social media could serve as a virtual focus group. The interaction of people in various ways across the world suggests that social media is a gold mine for understanding customer satisfaction and for identifying and suggesting new product features and services online. The reviews that customers leave are absolutely crucial when it comes to the profits and the reputations of the businesses and products associated with financial institutions. They often serve as the final quality-control check that a customer performs before making a purchase decision. This makes understanding the sentiment of the typical review very important, especially for those in positions of power making decisions that could affect the performance of the institution either positively or negatively. Earlier studies relied on human-annotated text to create LBMs for training ML models to perform SA. While this approach produced useful models, they were not truly exhaustive; they weren't very robust and certainly did not capture the entire range of structure and semantics in natural language. For our research, we are using advanced natural language processing for something LBMs do not handle well: comprehensively and accurately answering questions with a full range of possible subjects. We began by gathering customer reviews from Hellopeter, a well-known review platform, and financial information from the top five JSE-listed financial institutions in South Africa. Then, we employed OpenAI's ChatGPT as a zero-shot learning model to produce human-like annotations for several different sentiment tasks.

The feature vector from OpenAI ChatGPT was then input into BERT, BiLSTM, and a SoftMax function to measure and categorize the sentiment of a particular sentence.

We further added oversampling methods to the feature vectors to rectify the imbalanced data problem that we encountered and visualized the text and polarity of the customer reviews collected. The experiments showed that our method performed either as well as or better than the recent cutting-edge methods on the datasets we tested and yielded results that were comparable to them. The combination of OpenAI's ChatGPT with pre-trained BERT and BiLSTM models outperformed all other models and obtained an average score of 98.9%, an F1-measure of 97.7%, and an AUC of 91.90% when oversampling was used. The traditional model based on lexicons, SVMs, and LR achieved a score of 86.68% and an AUC of 91.90%. The study demonstrates the striking competence of OpenAI ChatGPT for accomplishing the tedious work of annotating customer reviews with emotional tone or polarity. The outcomes highlight the pronounced benefits of integrating customer SA with financial analysis and decision-making, which provides the organization with an additional understanding and prioritization of customer preferences and requirements. The study was particularly effective at parsing the model used in this research to understand and annotate customer questions and feedback (see Figure 4.1). In terms of theory and practice, this study adds a few significant dimensions.

- We introduce a technique specifically designed to collect data from the HelloPeter site and to conveniently arrange customer reviews for such tasks as SA. We use an efficient web scraping method to perform the task.
- We utilize OpenAI's ChatGPT to tackle the issue of subjectivity within the language and to classify the sentiment for each sentence. We use the zero-shot learning technique to label the data without prior training.
- We employ OpenAI's ChatGPT to comprehend the shape and meaning of customer reviews together with DL models to predict sentiment for individual sentences. To address the imbalanced dataset, we apply oversampling techniques.
- To enhance the explainability of the presented approach, an interactive system was developed to demonstrate the effect of sarcasm prediction on data acquisition and to compare performance experimentally with existing methods. The interactive system effectively demonstrates the worth of forecasting sarcasm in an online social media dataset and how AI technologies can be used to improve current techniques by precisely recognizing sarcastic remarks..

4.1 Input Layer

The input layer is made up of datasets that come from the HelloPeter website. HelloPeter is a one-of-a-kind consumer review platform based in South Africa that essentially serves as a user-generated evaluation space for a huge variety of businesses and services. These domestic critiques then form the input layer architecture of the

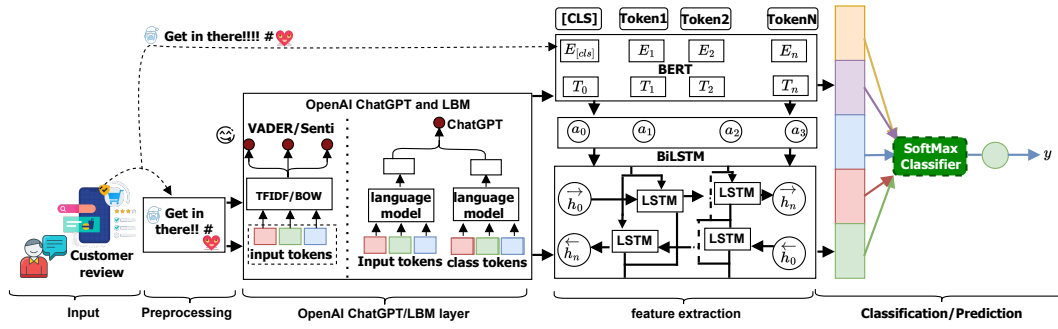


FIGURE 4.1: Structure of OpenAI ChatGPT BERT-BiLSTM network for SA.

proposed model as shown in Figure 4.1. In contrast to other review sites, HelloPeter focuses on domestic and specific sector evaluations that encompass a huge variety of industries—from banking and healthcare to hospitality and the tourism domain.

HelloPeter provides businesses with a means of monitoring and managing the discussion that occurs around their brand. Using HelloPeter, a company can not only get the rundown on how many positive and negative reviews it's garnering but can also dive deep into the details of the kinds of users that are leaving both forms of feedback. Companies benefit from these consultative features, and so do the users leaving reviews since they can now attach their reviews to HelloPeter's external social media sharing function.

The objective of this study is to apply OpenAI's ChatGPT, a zero-shot learning model, to the task of annotating customer feedback for its intrinsic subjectivity. We engaged ChatGPT to generate sentiment annotations using the natural language processing capabilities without any training specific to this task. We then processed those annotations through a hybrid DL model, BERT-BiLSTM, that integrates BERT's contextual understanding with the sequential information processing for which BiLSTM is well-known. The output of BERT-BiLSTM is the sentiment context of our dataset, annotated at the level of individual records.

4.1.1 Preprocessing

Businesses and organizations engage and check in with their customers through social media and various online platforms, like the HelloPeter website. In return, they receive valuable input from consumers that guides business decisions—thanks to SA. The feedback, however, comes with an obvious challenge: it's often drenched with personally identifiable information. Consumer feedback can cover a wide range of topics, from likes and dislikes to why a certain business establishment should be avoided at all costs. This makes preprocessing a multifaceted job in which several necessary tasks are woven together—protecting privacy, ensuring data integrity, and making the feedback truly informative.

The first step in preprocessing is to safeguard the privacy of individual reviewers while splitting the dataset into training and testing parts. Because of confidentiality concerns, we cannot provide personal identifiers like names, email addresses, or phone numbers in the datasets. All of that information has been removed. We made these decisions to stay well within the boundaries set by privacy laws and to work in an ethical and secure manner [14]. Of course, not all elements that might count as personal identifiers can be easily classified. To further clean the data, we also stripped out any superfluous HTML elements that might otherwise count as formatting noise.

To accomplish a more consistent standard for the text, accent characters are removed, and contractions like “can’t” are converted to their expanded forms, like “cannot”. The text is rendered completely in lowercase, special characters are eliminated, and any found symbols are replaced with alphabetic characters. The SpaCy library is used for the final step of lemmatization. This reduces words to their base or root forms so that we treat variations of the same word as the same word (e.g., “running” and “ran”). Running through these preprocessing steps gets us to the final, functional result that we literally look for in SA—that source material is as obfuscated as it needs to be without us losing any meaning from the text [130, 131].

4.1.2 OpenAI ChatGPT Layer

The capabilities of OpenAI’s ChatGPT are so advanced because of the processing power of the artificial intelligence that drives it. When we use it in this study, we do so with a complete understanding that this method differs from traditional ML that relies on matching predetermined words and phrases in order to predict what a piece of human writing might mean. In this study, we employ OpenAI’s ChatGPT, which is constructed upon the GPT-3.5 framework and utilizes the “text-davinci-003” embedding [132]. ChatGPT possesses a far more nuanced and accurate understanding of human sentiment because it has been trained on an almost enormous dataset compared to its predecessors [133].

A major step forward has been made in creating automated annotation tools with the emergence of ChatGPT. This zero-shot learning model requires no prior training to perform its tasks. The kinds of models that this new tool represents do not depend on huge quantities of well-labeled data, which past models needed to produce useful results. Previous attempts at automatic annotation required a lot of human-labeled data to work well, and they generally didn’t work very well. In contrast, ChatGPT and the models coming out of OpenAI these days seem capable of understanding and generating the sort of sentiment we want to be generated—without any human-labeled data to guide them.

The main goal of this layer is to utilize ChatGPT to generate human-like annotations for an array of text sources, including customer reviews, social media postings, and

news articles. This method confronts some of the well-documented limitations associated with traditional techniques, particularly those relying on lexical models. Using human feedback in a closed loop, we arrive at sentiment annotations that are not only more accurate but also more subtly interpretable. For each text or sentence x in our training data, we query ChatGPT for a sentiment score concerning the content in x . The score it gives then becomes the annotation for the full sentence.

We evaluate the performance of ChatGPT for annotation by comparing it with some of the best SA systems, including AFINN, SentiWordNet, and VADER. Section 4.2 provides a detailed description of these systems and of ChatGPT. Once we obtain the sentiment annotations from these models, we pass them to a BERT-BiLSTM layer, which extracts features that are useful for our model. Finally, a SoftMax layer delivers the sentiment classification of our sentiment text, making use of the AI-based programs as well as the traditional SA ones to achieve a simple, yet refined, classification system.

4.1.3 Zero-shot Learning with ChatGPT

While we investigated various methodologies for performing SA, we selected OpenAI's ChatGPT to actually carry out the analysis. Using the "*text-davinci-003*" model, which is an enhanced version of "*text-davinci-002*"—we sought to find a novel and efficient way to do SA. The "*text-davinci-003*" model had been enhanced using a well-known reinforcement learning strategy called Proximal Policy Optimization. More importantly, however, the model is an improved version of its predecessor primarily because it included RLHF, a key new training strategy that makes the model much better than previous ones. We updated and built upon the earlier work of SA studies to make them current and also integrated more recent works. We trained the model on all relevant tasks that are typically related to SA and are also used for NLP. We further investigated the ability of the model to complete these tasks when it is not given explicit previous examples to work from.

The progression of SA has reached a stage where it can not only classify the sentiment of a usual text but also handle texts of which it has no prior knowledge. When adopting this skill in text classification, one might term the process as zero-shot learning. In this study, we used ChatGPT, a model based on the 3.5 layer of the OpenAI GPT architecture, which has shown significant performance enhancements in SA tasks. Traditional models for SA typically require pre-training on particular datasets and perform sentiment tasks in a way that makes them seem anything but human. ChatGPT, in contrast, delivers human-like performance and is not fine-tuned for SA at all. To use ChatGPT for this purpose, we sent it "positive" and "negative" texts and captured the response of the text classifications performed by the model.

This study took sentences x from a training dataset and fed them to ChatGPT, with the goal of using the large language model to generate a sentiment score for the given content in the sentences x . The scores then served to annotate the training dataset. To check the validity of the scores that ChatGPT generated, the process we used could be likened to using a ruler to draw a straight line. The sentiment scores were generated and served as a kind of map for sentiment annotations. The dataset was then processed using BERT, which created word vectors dynamically from its pre-trained language model. These vectors were then input into a BiLSTM network that extracted sentiment features by grasping the contextual semantics encoded within the word vectors. At the end of the line, a SoftMax layer determined the sentiment polarity of the text in question with some degree of PR.

4.2 Lexicon-based Models

SA models based on lexicons begin with a precoded sentiment lexicon that contains a set of terms. These terms are associated with positive, negative, or neutral sentiment values, and they help to assign sentiment to an arbitrary piece of text [134]. The SA task is performed by matching words or phrases in the catchment area to entries in the sentiment lexicon. If a catchment area word has a sentiment lexicon entry, then the word is assigned that sentiment value. For our study, we employed three LBMs—AFINN [135], SentiWordNet [39], and VADER [11]—that approximate this SA approach as they perform the task of matching words or phrases in a catchment area to entries in a sentiment lexicon. The results of each model are discussed in the following subsections.

4.2.1 AFINN-Lexicon Model

AFINN's lexicon structure is a rule-based system that classifies sentiment according to rules that are set in advance [36]. But these rules don't come out of thin air; they're based on pretty smart statistical modeling. An important stop signal was found in this modeling: words have sentiment weights that vary depending on what context you're using them in [34, 35]. The AFINN lexicon comprises English words that have been manually assigned sentiment values. These values range from -5, which indicates a very negative sentiment, to +5, which denotes an extremely positive sentiment.

AFINN is well-known for being computationally efficient, making it particularly suited for analyzing large-scale datasets in environmental contexts where processing power may be constrained [37]. This approach's simplicity occurs alongside its effectiveness, allowing for rapidly reachable insights when large-scale textual data requires quick sentiment classification. AFINN's usage in SA tasks is prominent, rendering it perhaps one of the clearest choices of tool when SA task analysis requires a Luzern-class working model.

4.2.2 SentiWordNet Model

Our analysis employed the SentiWordNet model to attach sentiments to the words in our customer reviews. SentiWordNet assigns positive and negative scores to words based on what they mean in different contexts (their synsets) and based on which of their parts of speech (noun, verb, etc.) is dominant. We took these components and worked out what they meant in the context of our reviews [136, 137]. On the basis of what we extracted from the reviews, we then classified each review as either “positive” or “negative”.

SentiWordNet provided a more sophisticated and detailed analysis of the sentiment composition of the reviews compared to simpler LBMs [138]. The use of synsets allowed for a comprehensive understanding of the sentiment being expressed in the text with a nuanced comprehension that takes into account the various meanings and contexts in which words appear. This kind of analysis yields not only a breakdown of the sentiment along different axes but also an understanding of the review’s overall level of objectivity vs. subjectivity. When we integrated this model with our framework, we saw a distinct improvement in the SA we were performing on the HelloPeter reviews.

4.2.3 VADER

VADER is a rule-based SA tool that derives its power from iconic features that mimic human coding of sentiment. VADER assigns a polarity score to words in the lexicon. A word’s polarity score tells us how positive, negative, or neutral it is [13]. We used VADER because our review texts are relatively short, they express clear sentiment, and the lexicon of VADER is particularly good at coding the kinds of words one might use to express sentiment in a consumer review.

VADER combines grammatical rules with syntactical patterns to accurately detect sentiment in each review across the dataset [139, 140]. This means it looks at how sentences are put together to really understand what kind of sentiment is being expressed. Capturing what was previously thought to be just noise. VADER allows a more nuanced evaluation of sentiment and gives a clearer picture of what is being said (or not said) in a review. We used VADER to calculate a compound sentiment score for this study. The compound score looks at both the positive and negative sentiments expressed in a review. A score above 0.4 classifies a review as being positive, while a score below 0.4 classifies it as negative in sentiment [136, 137]. This study used VADER to provide nuanced SA of the customer reviews. VADER gave additional PR to the emotional tone of the reviews and enhanced the classification of the customer feedback.

4.3 Experiments

This section examines the experiments carried out on the suggested OpenAI ChatGPT BERT-BiLSTM system seen in Figure 4.1. We lay out the settings, which comprise an arrangement of experiments, to assess the model's performance. We likewise relate how the parameter settings were chosen, which assessment metrics we need, whether we're utilizing oversampling or not, and how our model's performance compares with some different tweaks of sentiment algorithms. Finally, we touch on the feature vector interpretation method we've settled on.

4.3.1 Experiments Setup

This study ran proposed methods on a personal computer that operated Ubuntu 20.04 and were equipped with a 4.0 GHz Intel Core i7 and 64 GB of DDR4 memory. The tests performed in this study utilized the sci-kit-learn library in conjunction with Python 3.7 and Keras, which was backed by TensorFlow 2.0.

4.4 Evaluation Metrics

We assessed the performance of the OpenAI ChatGPT BERT-BiLSTM system we constructed as illustrated in Figure 4.1. It was evaluated using some clear, key performance metrics such as ACC, F-measure, and AUC. These provide a pretty complete picture of the model's performance, especially when the true labels are not evenly distributed, as is often the case with our datasets. Finally, we built a confusion matrix to get another look at how well the model performed on our test set, which it had never seen before.

4.4.1 Accuracy

The ACC is a basic measure that counts how many instances the model predicted correctly (both the true positives and the true negatives) out of all the instances it predicted. It is defined mathematically as shown in Equation 4.1:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

Where:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

4.4.2 F-measure

The F-measure gives a balanced estimate of a model's performance, taking into account both PR and RC. PR is the ratio of the number of positive instances that the model predicts correctly to the number it predicts at all. RC is often viewed as a complementary measure; it indicates how well a model can find the positive instances it is supposed to find when it is applied to an ensemble of instances in which some are known to be positive. The F-measure is then computed as the harmonic mean of PR and RC, as shown in Equations 4.2, 4.3 and 4.4 :

$$PR = \frac{TP}{TP + FP} \quad (4.2)$$

$$RC = \frac{TP}{TP + FN} \quad (4.3)$$

$$F - Measure = 2 \cdot \frac{PR \cdot RC}{PR + RC} \quad (4.4)$$

The F-Measure is very beneficial when working with imbalanced datasets. It places value on PR, RC, and, most importantly, the balance between the two. This makes the F-Measure an effective metric for evaluating the performance of a model when using an imbalanced dataset.

4.4.3 Area Under the ROC Curve

A binary classification model's performance can be reliably assessed using the area under the ROC curve. The ROC curve itself plots the model's true positive rate against the false positive rate over a range of probability thresholds. A higher AUC value is associated with a model that better separates the positives from the negatives. And it is particularly useful as a metric when one has a classification problem with a skewed dataset or when the misclassification costs are different between the two classes.

The AUC is It is defined mathematically as shown in Equation 4.5:

$$AUC = \int_0^1 \left(\frac{TP}{P} \right) d \left(\frac{FP}{N} \right) \quad (4.5)$$

Where:

$$P = TP + FN \quad (\text{Total number of positives})$$

$$N = TN + FP \quad (\text{Total number of negatives})$$

This can also be expressed as the integral of the TPR with respect to the FPR as shown in Equation 4.6:

$$AUC = \int_0^1 TPR \, dFPR \quad (4.6)$$

Where:

$$TPR = \frac{TP}{P}$$

$$FPR = \frac{FP}{N}$$

Alternatively, the AUC can be written as shown in Equation 4.7:

$$AUC = \frac{1}{P \times N} \int_0^N TP \, dFP \quad (4.7)$$

This expression underscores that when one calculates the AUC, one is effectively integrating the true positives across various counts of false positives. Counting true positives in this way gives a performance measure for the classifier. A random classifier has an AUC of 0.5, while a perfect classifier achieves an AUC of 1.

4.4.4 Evaluation

In this section, the results, which we have obtained with the help of the BERT-BiLSTM system of the OpenAI ChatGPT model, are presented in the figure below; see Figure 4.1. We observed that the dataset of customer reviews collected from HelloPeter contains reviews and has the unique feature of a highly imbalanced class distribution, which is clearly shown in Figures 4.2 and 4.3. In opposition to this, we employed methods based on Lexicons, as detailed in Section 4.4, in which features were generated and ML classifiers trained with the help of ChatGPT. That is why this process helped to enrich and expand the knowledge concerning the materials of the review data received. The sentiment distribution analysis we have done on the ML classifiers' prediction was at the sentence level. The sentiment label detected by the two ML classifiers is also indicated for each sentence. These results were then accumulated, and for each model, a distribution of the sentiment labels that were assigned was generated. The overall results are summarized in Table 2.2.

The association between the syntactic structure of a sentence and its baseline subjectivity in natural language is underscored by the results in Table 4.1. To boost the performance level of our sentiment-analysis model, we fused LBMs like AFINN and SentiWordNet with a rule-based algorithm known as VADER. Additionally, we used ChatGPT to help generate some of the features for our model. Our dataset, which

is imbalanced, requires that we use a technique to boost the underrepresented features. We applied the SMOTE technique for this purpose, as discussed in Subsection 4.5.1, to artificially augment the minority class.

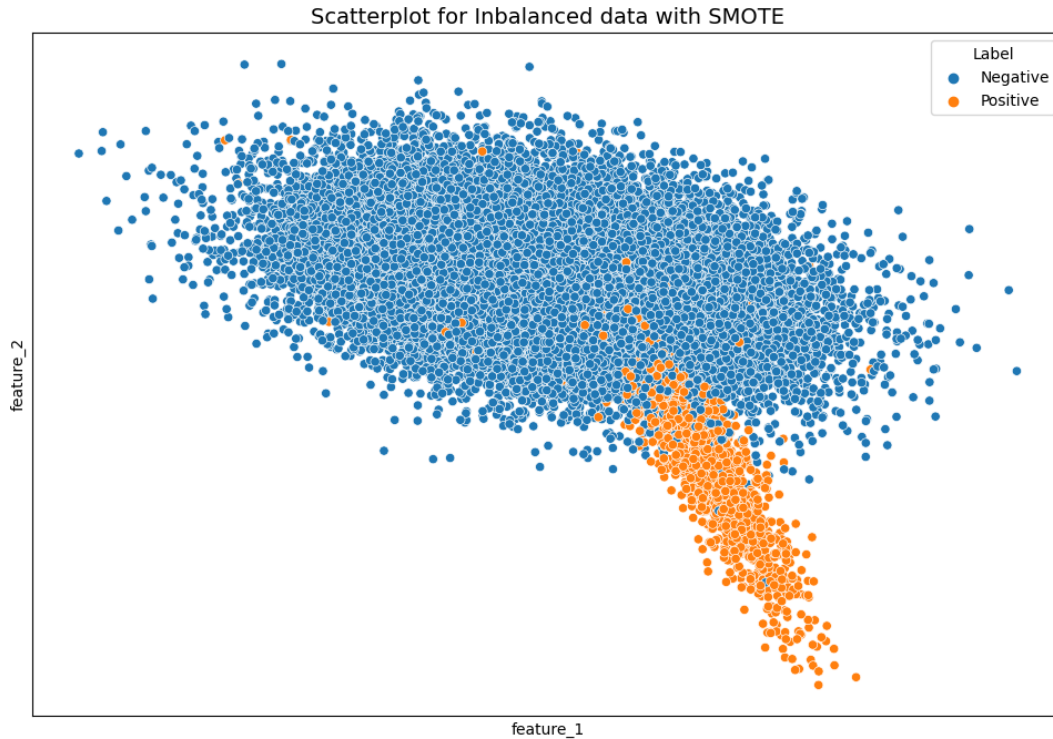


FIGURE 4.2: 2-D feature distribution without SMOTE

TABLE 4.1: Summary of class distributions for the combined dataset.

Model	Total Size	Positive	Negative
AFINN	129,341	34,275	95,066
SentiWordNet	129,341	61,049	68,292
VADER	129,341	54,849	74,492
OpenAI	129,341	13,315	116,026

4.5 Data Augmentation

Data augmentation techniques are the different ways to enhance the diversity and quantity of training data without going out to collect new data. They are particularly useful for unbalanced datasets like ours, where one sentiment class has many fewer examples than the other class. This section looks at SMOTE, one prominent way to generate new examples in the underrepresented class. SMOTE works right in the data space of our unbalanced datasets and doesn't make any assumptions about our model's architecture or the kinds of data it will be seeing when it makes its sentiment predictions.

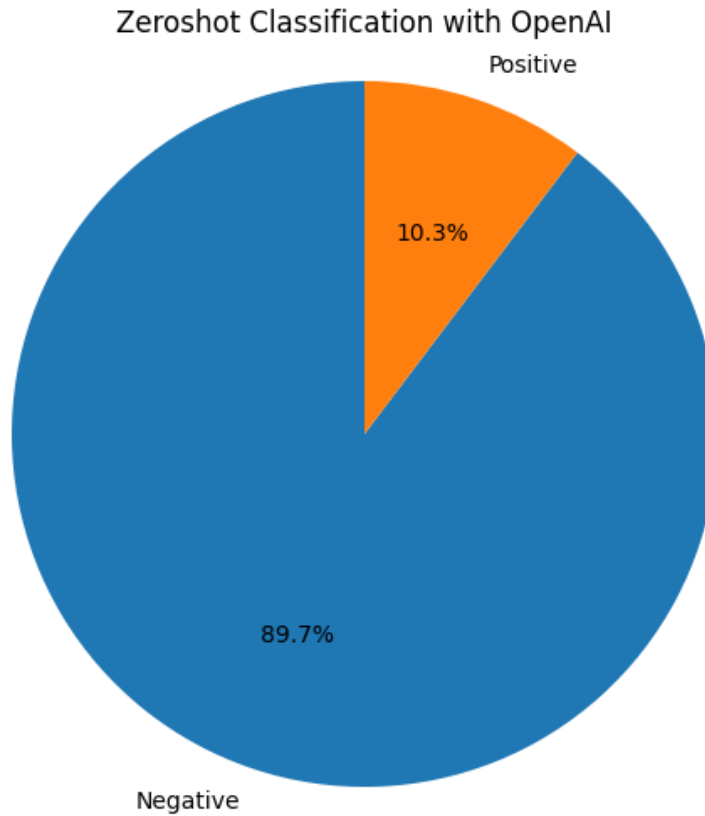


FIGURE 4.3: ChatGPT OpenAI feature distribution without SMOTE

4.5.1 Synthetic Minority Over-sampling Technique (SMOTE)

To solve the problem of class imbalance in our datasets, we used an important tool known as SMOTE. SMOTE allows us to do some necessary oversampling on our minority class. Even though the SA problem is very intensely studied, it remains a problem that researchers are grappling with. In our case, the class we are trying to identify as “positive” in a given text is our minority class. We worked with the oversampled data, and this allowed us to achieve much better performance in terms of both model ACC and interpretability.

We used various oversampling techniques to create a proportionate and balanced dataset and then classified and evaluated the model using the sklearn library in Python¹ [141]. Among these techniques, SMOTE was critical. It allowed us to generate in-depth synthetic samples of the minority class. This was essential to shoring up the data as we headed into the modeling stage. SMOTE seemed to be working magic in ensuring that we were not biased towards the majority class. The class balance in the dataset ensures fairer training of the models, improving their performance in predicting sentiment, as shown in Algorithm 2. Figures 4.4 and 4.5 show the label distribution and the class balance achieved using SMOTE in the 2-D feature space, as well as the feature distribution from ChatGPT.

¹<https://scikit-learn.org/stable/>

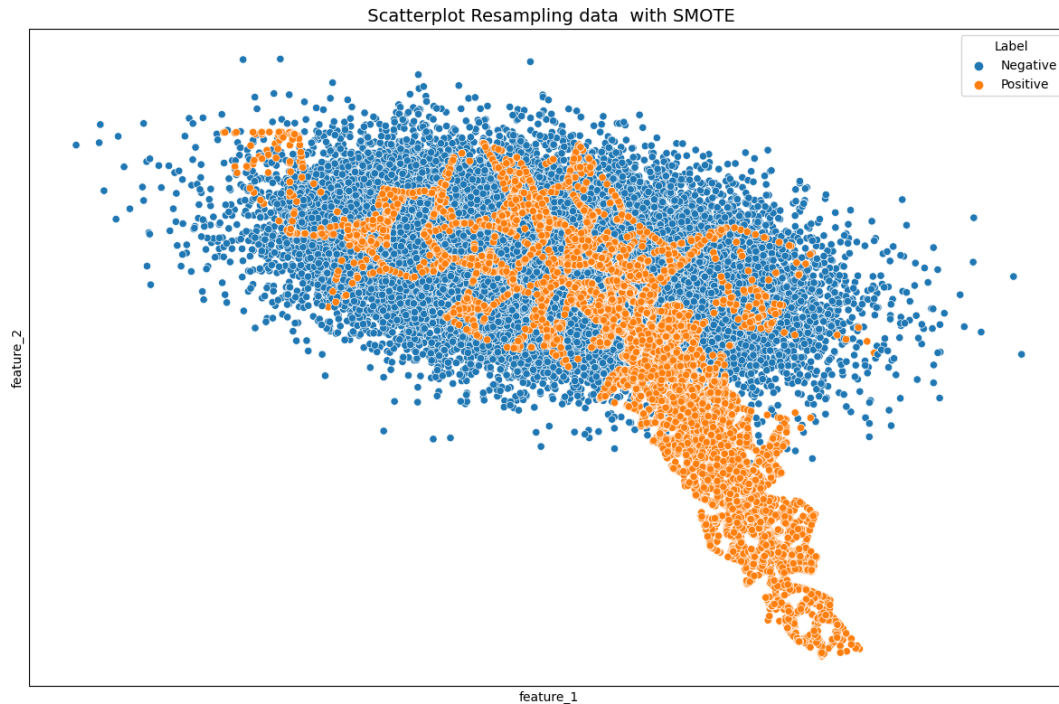


FIGURE 4.4: 2-D feature distribution with SMOTE

We assessed how well SMOTE worked at improving the ACC of our SA models. For this task, we treated the dataset as if it were a binary classification problem and utilized a RF algorithm for classification. Among the many ways one could use RF, we chose to implement repeated stratified k-fold cross-validation (with $k = 5$) to get a handle on the ACC of our F-scores and, by extension, the RC rates. Figure 4.6 shows the performance of several models trained on various iterations of class-balanced data.

The experimental findings illustrate that SMOTE was a significant enhancement, or noticeable improvement, in achieving RC and the F-measure. When we began, our target was to achieve consistent RC values as depicted in Figure 4.6. While we did not quite get there, we obtained some very consistent return values with SMOTE, and as our target service layers moved further away with 50:50 distribution across the 2 classes, we obtained better F-measure values, which is perfect when it comes to multi-class problems. As seen in Table 4.1, the balanced dataset resulted in improved sentiment prediction ACC.

The results imply that the overall performance of SA models is better when the datasets are balanced, and this is done by a particular type of synthetic oversampling called SMOTE. With the data balanced, sentiment models with no other changes made to them can achieve better results on some evaluation metrics used. From this outcome, we can take this as an indication that SMOTE, when used in this way, can yield even better results for SA model performance evaluations.

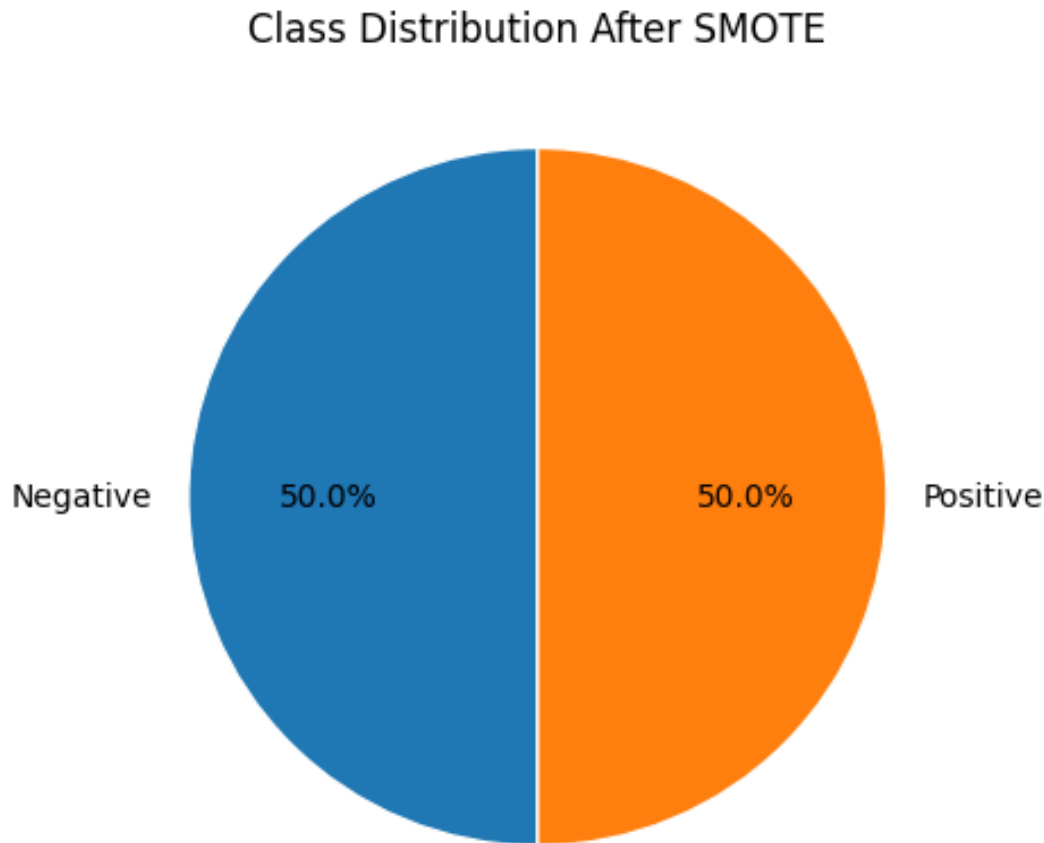


FIGURE 4.5: ChatGPT OpenAI feature distribution with SMOTE

4.6 Results

In this part, we give the specifics of how the model we proposed in Figure 4.1 carries out its function, both with and without the use of the oversampling strategy. This gives us the opportunity to hold a review and make comparisons that go deeper than a mere trivial assessment. The results we obtained from the dataset provide measures of ACC, F-measure, and AUC, and you can find them in Table 4.2.

The first portion of Table 4.2 presents the performance metrics of the SVM classifiers and the ChatGPT OpenAI feature selection method on customer reviews obtained for ABG on Hellopeter. The SVM classifiers had an ACC of 0.970 and an AUC of 0.976, showing that they are superior to the LR classifiers, which had the same ACC but an AUC of 0.971. The succeeding portion of the table indicates the performance of the SVM classifiers on AUCs; they had an ACC of 0.956 and an AUC of 0.940 in this instance. Comparatively, this suggests that the SVM classifiers are not as effective as the RF classifiers, which had an ACC of 0.920 and an AUC of 0.970, with roughly imbalanced datasets.

The second half of Table 4.2 demonstrates that the SVM model achieved the highest

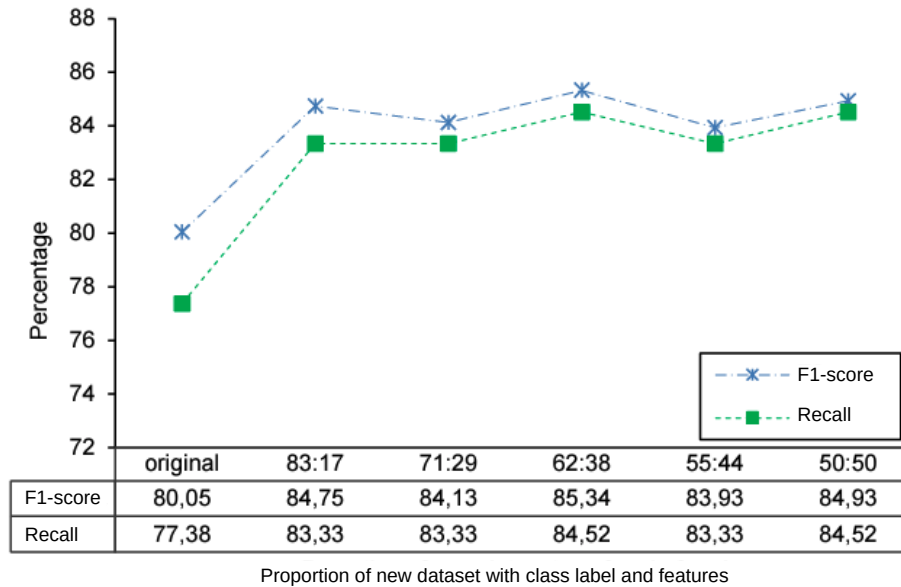


FIGURE 4.6: Performance metrics on the balanced dataset using SMOTE and RF.

values for ACC, F-Measure, and AUC for the CPI dataset taken from Helloworld, and this was without the aid of SMOTE. The actual numbers for these performances were as follows: for ACC, an outstanding 0.981; for F1-Measure, 0.93; and AUC, a good 0.906. Once again, when pitted against the RF model, the SVM came out on top for these same three measurements. When comparing with LR, while SVM had similar performances with them for F1-Measure and AUC, the ACC of the SVM was inferior, but not by too much, making the RF the most balanced model of the four. However, with the recent implementation of SMOTE, the CPI dataset in terms of ACC, F1-Measure, and AUC, which were 0.97, 0.96, and 0.99, respectively.

Furthermore, not only did the SVM classifier with the SMOTE technique show an increase over LR, but it also showed increased performance over the K-nearest neighbor, MNB, and RF algorithms when they were deployed on the standard dataset as shown in Table 4.2. In fact, the SVM classifier with SMOTE produced results that almost approached perfect performance with an ACC of 0.967, an F1-Measure of 0.957, and an AUC of 0.99. On the same standard dataset, the ChatGPT model achieved a very similar result: an ACC of 0.970. Yet while ChatGPT also produced an impressive performance that puts it near the semantic peak, we must remember that it did all of this without the benefits of SMOTE with an AUC of 0.975. Thus, it might actually make more contextual sense that SVMs + SMOTE are actually more understandable given their slight edge in F1-Measure results over ChatGPT.

The ACC of the SVM classifier and the score from SMOTE for the NED dataset were 0.974, 0.964, and 0.738 for ACC, F1-Measure, and AUC. This demonstrated

Algorithm 2 Pseudo-code of SMOTE Technique

Input: T : Number of minority class samples, N : Amount of SMOTE (percentage), k : Number of nearest neighbors

Output: Synthetic minority class samples

```

1: if  $N < 100$  then
2:   Randomize the  $T$  minority class samples
3:    $T \leftarrow \left(\frac{N}{100}\right) \times T$ 
4:    $N \leftarrow 100$ 
5:  $N \leftarrow (\text{int}) \left(\frac{N}{100}\right)$ 
6:  $\text{numattrs} \leftarrow$  Number of attributes
7:  $\text{Sample}[][] \leftarrow$  Array for original minority class samples
8:  $\text{newindex} \leftarrow 0$ 
9:  $\text{Synthetic}[][] \leftarrow$  Array for synthetic samples
10: for  $i \leftarrow 1$   $T$  do
11:   Compute  $k$  nearest neighbors for  $i$ , and save the indices in  $\text{nnarray}$ 
12:   POPULATE( $N, i, \text{nnarray}$ )
   FMainPopulate FnFunction:  $N, i, \text{nnarray}$ 
13: while  $N \neq 0$  do
14:   Choose a random number between 1 and  $k$ , call it  $\text{nn}$ 
15:   for  $\text{attr} \leftarrow 1$   $\text{numattrs}$  do
16:     Compute:  $\text{dif} \leftarrow \text{Sample}[\text{nnarray}[\text{nn}]][\text{attr}] - \text{Sample}[i][\text{attr}]$ 
17:     Compute:  $\text{gap} \leftarrow$  random number between 0 and 1
18:      $\text{Synthetic}[\text{newindex}][\text{attr}] \leftarrow \text{Sample}[i][\text{attr}] + \text{gap} \times \text{dif}$ 
19:    $\text{newindex} \leftarrow \text{newindex} + 1$ 
20:    $N \leftarrow N - 1$ 
21: return Synthetic

```

that when ChatGPT was used for SA, it was indeed the most effective and accurate method. Without SMOTE, ChatGPT achieved 0.981 for ACC, 0.873 for F1-Measure, and 0.740 for AUC. Conversely, when traditional methods like AFINN, SentiWordNet, and VADER were used and using a similar SMOTE technique, the ACC dropped to 0.821, 0.964, and 0.624 across these models when implementing SVM. Comparatively, K-nearest neighbor, LR, MNB, and RF fell behind with 0.594, 0.882, 0.738, and 0.738 ACC, respectively, on the AFINN model, which remains an unsatisfactory result for classifier sentiment text across the remainder of the standard as shown in Table 4.2.

Using the SMOTE technique, the SVM classifier ranks the best on the FSR dataset. Its scores for ACC and F1-Measure are quite impressive, coming in at 0.994 and 0.954, respectively. However, in regards to AUC, it doesn't look quite as strong, earning a score of 0.730. To put that in perspective, LR earned an AUC score of 0.993, which is the best score of this metric on the FSR dataset. When looking at AUC in terms of classifiers in ranked order, ChatGPT ranked RF third with an AUC score of 0.985; then came SVM in fourth with 0.804; and RF, MNB, and K-nearest neighbor were ranked in positions three through five from ChatGPT. The best F-Measure without SMOTE was achieved by LR, which obtained values of 0.917; RF

came in second place with 0.876; SVM took third with 0.874; while MNB and K-nearest neighbor occupied a tie for second/third place with 0.874. The highest ACC for LR, with AFINN as the SA tool, was 0.917 with SMOTE applied, followed by K-nearest neighbor with 0.876; SVM with 0.818; MNB with 0.778; and RF in the lowest place with 0.651. Simultaneously, SentiWordNet combined with K-nearest neighbors yields the highest ACC measure of 0.697 for VADER, followed by an SVM combination that scores 0.612 when SMOTE is used. In contrast, when ChatGPT is used as a classifier with the dataset FSR, a condition of the SVM combination yields the highest overall ACC measure of 0.994. However, the measure of performance known in the field as AUC is not as high as that in a few conditions, with the ABG dataset's highest being 0.975 as shown in Table 4.2. Finally, AFINN yields better ACC than the SMOTE model overall, with a K-nearest neighbors AUC of 0.996 in the mixed dataset.

Additionally, we improve the reliability of the proposed model, e.g., ChatGPT OpenAI for text annotation tools and BERT-BiLSTM to improve SA tasks. We implement cross-validation to fit the model and estimate the prediction ACC of the unseen sets of data drawn from the training samples. In doing this, we partition the data into two independent sets, one for training and one for testing, and then estimate the ACC of the events and tasks. The first set is used to train the model, while the subsequent set of data (the test) is used to evaluate its performance using a 10-fold cross-validation [142]. Cross-validation is a kind of resampling process that assesses the model and trains it roundly many times using test data from samples drawn from the original. In this paper, we used 5-fold stratified cross-validation with shuffling and a random seed for reproducibility. The results are shown in Table 4.3, which provide valuable information into the proposed model's Figure 4.1 performance and ACC. This process enables us to measure the model's generalizability to new data and identify any potential overfitting issues, which ensures that the proposed model is robustly constructed and reliable in predicting sentiment in unseen text data. Our test shows that ChatGPT OpenAI with the BERT-BiLSTM model and SMOTE technique is more accurate than the same model without the SMOTE technique. It reached scores of 0.932 for the FSR dataset, 0.922 for NED, 0.902 for STD, 0.885 for CAP, and 0.884 for ABG. The results highlight the effectiveness of using SMOTE techniques in improving model performance on imbalanced text datasets. In contrast, the result in terms of the AUC score also indicated that the FSR dataset outperformed the other datasets with an AUC score of 0.923, while the ABG dataset ranked second with an AUC score of 0.890, and the STD, CAP, and NED datasets came in third, fourth, and fifth, respectively, with an AUC score of 0.887, 0.883, and 0.880. Whereas when the ChatGPT OpenAI BERT-BiLSTM model was used without SMOTE technique, it did not perform at all as planned. For the ABG, CAP, STD, FSR, and NED datasets, the ACC and AUC scores were 0.505, 0.659, 0.786, 0.788, and 0.88, respectively. We add SoftMax as an activation function in the layer of the proposed OpenAI ChatGPT BERT-BiLSTM system in Figure 4.1 to normalise the raw

model outputs to ensure that the probability scores for each sentiment polarity sum up to one. This helps the OpenAI ChatGPT BERT-BiLSTM system shown in Figure 4.1 sort sentiments into groups based on their orientation. The high ACC and AUC scores in the evaluation section show how this approach helps to accurately classify sentiment orientations across datasets. This demonstrates the effectiveness of our model in predicting sentiment in unseen text data. This technique enables our model to offer probabilities that can be readily interpreted as confidence levels for each sentiment category.

In summary, the proposed model ChatGPT OpenAI BERT-BiLSTM with the SMOTE technique outperformed the standard models that use zero-shot learning in terms of ACC and AUC for the FSR dataset and the NED, STD, CAP, and ABG datasets when PR is taken into account. This speaks to the value of using pre-trained language models to do SA. For example, even though ACC is terms of aspect-based SA, for example, VADER does much better and is more accurate than it. 0.94 on the NED, only SentiwordNet SVM plus SMOTE does better than Lexicon rules and regular ML models that use them. In terms of aspect-based SA, for example, VADER does much better and is more accurate than it. However, when it comes to text annotation using a pre-trained language model, ChatGPT with its user-friendly interactive interface offers up something not available for the tools listed above.

4.6.1 Visualised design

In this section, we facilitate understanding of the OpenAI ChatGPT BERT-BiLSTM system, as depicted in Figure 4.1, for our non-expert users. The ChatGPT system conducts SA on customer review comments collected from the Hellopeter website and a variety of social media platforms across the Internet. These reviews relate to the product or service. We evaluated the proposed system against all of the traditional methods, with and without synthetic data samples generated via SMOTE, to see which way performed best in terms of ACC and efficiency and to identify which manner of conducting the analysis led us to the best interpretation of the found data.

The visuals in Figure 4.7 serves as alternative means of expressing the analysis from which the ML model was derived. That model now runs on a high-performance computer and was trained to detect and classify the sentiment expressed in each of the individual sentences found in the thousands upon thousands of customer reviews of all the banks that we collected. What these specific visualizations show is how well different ML algorithms carried out the SA task—an analysis that was performed concurrently with the main analysis of the training results for the ML models. And a few things really do stand out in this comparative analysis.

4.6.2 Confusion matrix

To better understand and analyze the classification of subjects into correct and incorrect categories in the datasets, we examine the confusion matrix. This analysis applies both to datasets preprocessed with the SMOTE technique and those that were not. By applying the analysis to both types of datasets, we can better assess the model's performance in predicting sentiment. The correct and incorrect confusion matrices are shown in Figures 4.8–4.12. They reveal the number of errors that were made as compared to the number of true positives for each class in all the datasets. The figures represent the confusion that occurs between two classes (positive and negative), which results in the most false positives and false negatives. The matrices also show us confusing errors between groups of classes when sentences from one class are mistaken for another class. In short, they show who is in the class, who is out of the class, and which classes are getting confused.

How information or datasets are compromised when the true class label is not indicated can be crucial for applications that use the data. In the absence of any other form of optimization, the predictions made by the OpenAI BERT-BiLSTM model for most classes appear to be on the diagonal in Figures 4.8–4.12. The diagonal represents the desired outcome of predicted class equaling actual class. The problem is that with no SMOTE, there's a substantial discrepancy between the predicted class and the actual class that affects the model in a negative way. The AUC performance observed in the path from ChatGPT OpenAI BERT-BiLSTM with SMOTE data is lower than that from ChatGPT OpenAI BERT-BiLSTM to AUC performance without the use of SMOTE, as shown in Figure 4.13. However, this path still provides better AUC performance across all datasets than the path without SMOTE. The results from the ChatGPT OpenAI BERT-BiLSTM model that processes customer reviews and employs the SMOTE technique across all datasets can be summarized as shown in Figure 4.14. The figure indicates the performance of this particular model in terms of several evaluation metrics that have been the subject of much discussion in previous sections.

4.6.3 Interpretation

The results are discussed in this section, along with what they can tell us about the feature vectors computed by our new SA method. They might help decision-makers be less subjective when it comes to SA tasks. We employed two different model-agnostic approaches for generating explanations for the outputs of our SA method.

The first approach is SHAP, or Shapley Additive Explanations, which uses cooperative game theory to explain the output of any ML model [143]. The SHAP value for a feature i is defined as 4.8:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (4.8)$$

where ϕ_i represents the contribution of feature i , N is the set of all features, S is a subset excluding i , and $f(S)$ is the model's output when only features in S are considered.

The second approach is LIME, or Local Interpretable Model-Agnostic Explanations, which is another framework for producing rationales for model outputs from a different perspective [144]. LIME learns an interpretable model g locally around an instance x' by solving as 4.9:

$$\arg \min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \quad (4.9)$$

where G is the family of interpretable models, g is the selected interpretable model, $\mathcal{L}(f, g, \pi_x)$ is the loss function measuring how well g approximates f locally, π_x is a proximity measure assigning higher weights to samples closer to x , and $\Omega(g)$ is a complexity penalty to ensure interpretability.

These methods provide insights into how the SA method makes predictions, improving transparency and trust in the results.

Figure 4.15 shows the predictions from randomly selected sampled documents in the FSR datasets. As shown in Figure 4.15, the proposed method puts more weight on the positive label, with a significant prediction probability on words such as appreciation at 7%, save at 7%, and excellent at 7%, with a phrase such as "appreciation post received excellent service karabo aphone branch paled mall thank much karabo able to save money information provided" This shows that 100% of the text in this phrase is coming from a positive class label. Equally, with another phrase within the same FSR dataset, we have a significant prediction probability on words such as debt at 36%, destroy at 22%, and steal at 22%, and within the same phrase, words like please and free are labeled as positive with 19% and 17% predicted within the sample of the text, while the entire perception of the phrase reaches 99% negative and 1% positive, respectively.

Another scenario shows the proposed model through LIME highlights words such as "rude" at 60% compared to the words "horrible" and "arrogant" at 19% and 18% (as a feature now) to classify the text sample as 77% negative and 23% positive sentiment, which is the confidence the proposed model has in the text. As Figure 4.15 demonstrated, it provides non-experts with a great way of elucidating what is going on when training the ChatGPT Open BERT-BiLSTM with SMOTE techniques on FSR datasets for non-technical folks rather than just being a black box that can not be interpreted.

Additionally, we also applied SHAP [144] to the feature vectors to get a good estimate of the SHAP values for the extracted features. The permutation was accomplished by turning one word on and off. For example, if a word is missing from a particular post, enabling it could make a significant difference in revealing the inherent subjectivity of each sentence in the customer review in natural language, improving interpretation and explainability to non-experts in the field about what the proposed model is learning for effective sentiment classification of a given text. The document's nonexistent feature could obtain a high SHAP score. Before SoftMax, the collected features were subject to permutations to avoid this. By doing so, the proposed model was able to better understand the importance of each feature in the sentiment classification process. This approach helped ensure that the model's predictions were based on meaningful and relevant information.

Figures 4.16 and 4.17 depict the first random sample from the FSR databases. The SHAP scores for the contributing characteristics picked during the training set of the proposed model are also displayed. The features that push the prediction higher are shown in red, representing the corrected prediction by the model used in predicting and classifying a sample text as positive, and those that push the prediction lower (predicting the incorrect as negative) are shown in blue, along with the confidence level. The SHAP local explanation takes only one occurrence at a time and creates an explanation by indicating which feature values make decisions about the position and which are negative. Figure 4.16 and 4.17 demonstrates the robustness of using SHAP to explain the complexity of the proposed model with the customer reviews dataset to understand and analyse the subject in each sentence and make it explainable. We observe in Figure 4.16 and 4.17 that the probability of the proposed model confidently predicting the sample text based on the characteristics highlighted in red is 48.22% as a positive sentiment, which includes words like "take care", "loyalty", and the likelihood of using features in blue as negative is 1.44% using the words like "pay debt", "including debt", and "get punished kid" using the FSR dataset, where based probability confident of using words like "assign incident", "frustrated" are tailored towards a negative in blue at 0.12% and positive sentiment at 8.61%. These wholeheartedly correlate to the same explainable interpretation using LIME in Figure 4.15.

We will also look into how the SHAP model can be used to enumerate explainable components of specific sentences from FSR customer review documents. The idea here is that SHAP takes the input features, sums up the difference between the baseline (expected or value) model output, and uses the results as the current model output. The easiest way to implement this in SHAP is to use a waterfall plot that starts with our background expectations of the model as a function to find the inherent subjective value of a given sentence as $[f(X)]$, and then you can start adding features one at a time until we reach the current model output.

4.7 Summary

The chapter described ChatGPT as a tool for text annotation that can analyze sentiment. In traditional SA pipelines, this task would be performed by LBMs such as AFINN, SentiWordNet, and VADER. While these methods have proven effective, they also have limitations. ChatGPT, used in a zero-shot manner, provides an excellent alternative for classifying sentiment when LBMs fall short. A zero-shot pipeline requires no pre-labeled training data. Furthermore, we showed how to apply SMOTE, another text annotation tool that ChatGPT complements, across the same dataset with the following key improvements resulting: performance at the level of ACC, PR, RC, and F-measure. Finally, we stated explicitly that using SMOTE while accounting for the kind of class imbalance present in our dataset led to these key improvements.

A simultaneous investigation into ChatGPT and LBMs yielded some very illuminating results regarding the characteristics of each model. Given ChatGPT's impressive generative capabilities and the overall human-like quality of its text, one might expect it to come out on top in sentiment classification tasks. But in our studies, it frequently came in a distant second (or even third, behind some variants of LBMs). The reason seems to be that these LBMs are not just good but excel at tasks with a clear type of structure—e.g., discerning between positive and negative sentiments. To sum up the chapter: Models like ChatGPT are best used in conjunction with and on top of the traditional techniques that SA researchers have been using for years. These techniques balance out the very unbalanced class distributions that many state-of-the-art models have a problem with. By addressing class imbalances, the following chapter will delve into the use of retrieval-augmented generation (RAG) methods to enhance sentiment analysis through improved data retrieval and model performance

TABLE 4.2: ACC, F-measure, and AUC results for the proposed model

ABG															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.651	0.587	0.696	0.659	0.628	0.733	0.731	0.618	0.626	0.706	0.589	0.696	0.709	0.588	0.697
SentiWordNet	0.686	0.683	0.502	0.693	0.691	0.508	0.552	0.624	0.695	0.685	0.640	0.645	0.524	0.760	0.694
VADER	0.551	0.541	0.518	0.507	0.502	0.692	0.563	0.615	0.510	0.571	0.643	0.697	0.581	0.628	0.500
ChatGPT OpenAI	0.908	0.871	0.731	0.970	0.873	0.976	0.971	0.912	0.971	0.961	0.871	0.610	0.887	0.877	0.806
ABG with SMOTE															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.501	0.734	0.510	0.819	0.817	0.976	0.933	0.932	0.747	0.754	0.640	0.900	0.650	0.737	0.696
SentiWordNet	0.714	0.767	0.738	0.825	0.923	0.951	0.817	0.808	0.849	0.826	0.722	0.715	0.812	0.611	0.614
VADER	0.703	0.840	0.914	0.909	0.906	0.733	0.864	0.958	0.818	0.921	0.760	0.885	0.966	0.794	0.758
ChatGPT OpenAI	0.502	0.737	0.527	0.956	0.956	0.939	0.949	0.723	0.821	0.920	0.920	0.993	0.916	0.915	0.967
CPI															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.681	0.640	0.570	0.674	0.620	0.556	0.695	0.571	0.512	0.727	0.612	0.521	0.714	0.595	0.699
SentiWordNet	0.678	0.675	0.673	0.693	0.693	0.546	0.567	0.547	0.529	0.535	0.683	0.503	0.518	0.754	0.688
VADER	0.525	0.505	0.688	0.507	0.699	0.602	0.559	0.609	0.695	0.582	0.647	0.687	0.581	0.627	0.525
ChatGPT OpenAI	0.911	0.896	0.928	0.981	0.930	0.906	0.974	0.924	0.922	0.962	0.561	0.798	0.889	0.777	0.608
CPI with SMOTE															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.602	0.519	0.525	0.826	0.824	0.932	0.855	0.849	0.931	0.792	0.787	0.711	0.659	0.650	0.531
SentiWordNet	0.501	0.735	0.545	0.520	0.519	0.576	0.514	0.698	0.638	0.584	0.582	0.507	0.503	0.502	0.525
VADER	0.693	0.672	0.525	0.615	0.612	0.672	0.613	0.602	0.720	0.642	0.638	0.597	0.568	0.562	0.931
ChatGPT OpenAI	0.501	0.736	0.531	0.974	0.964	0.992	0.920	0.910	0.807	0.910	0.727	0.845	0.901	0.835	0.876
SBK															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.628	0.600	0.698	0.661	0.620	0.667	0.733	0.621	0.693	0.708	0.588	0.504	0.709	0.588	0.547
SentiWordNet	0.504	0.502	0.698	0.507	0.506	0.634	0.532	0.787	0.530	0.544	0.513	0.692	0.524	0.760	0.660
VADER	0.693	0.684	0.660	0.694	0.689	0.641	0.559	0.612	0.688	0.569	0.648	0.504	0.581	0.628	0.511
ChatGPT OpenAI	0.915	0.877	0.547	0.970	0.874	0.694	0.975	0.918	0.665	0.962	0.871	0.698	0.887	0.877	0.698
SBK with SMOTE															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.543	0.657	0.511	0.811	0.809	0.974	0.910	0.909	0.899	0.764	0.759	0.706	0.652	0.645	0.530
SentiWordNet	0.514	0.764	0.541	0.543	0.541	0.667	0.600	0.592	0.627	0.595	0.592	0.508	0.507	0.506	0.683
VADER	0.595	0.647	0.683	0.602	0.593	0.707	0.657	0.598	0.694	0.620	0.647	0.594	0.568	0.616	0.561
ChatGPT OpenAI	0.522	0.735	0.530	0.967	0.957	0.992	0.951	0.992	0.809	0.921	0.910	0.910	0.913	0.730	0.764
NED															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.657	0.611	0.532	0.654	0.605	0.678	0.708	0.587	0.654	0.714	0.594	0.511	0.709	0.588	0.630
SentiWordNet	0.515	0.509	0.696	0.697	0.696	0.509	0.515	0.505	0.695	0.505	0.667	0.694	0.517	0.755	0.678
VADER	0.501	0.693	0.678	0.523	0.518	0.611	0.548	0.605	0.511	0.587	0.655	0.681	0.580	0.626	0.506
ChatGPT OpenAI	0.907	0.868	0.630	0.982	0.873	0.740	0.977	0.921	0.635	0.969	0.865	0.676	0.887	0.878	0.506
NED with SMOTE															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.593	0.652	0.500	0.821	0.820	0.877	0.882	0.877	0.970	0.739	0.729	0.916	0.739	0.729	0.916
SentiWordNet	0.696	0.733	0.522	0.964	0.964	0.715	0.908	0.907	0.996	0.744	0.740	0.824	0.558	0.652	0.601
VADER	0.500	0.518	0.580	0.624	0.622	0.744	0.607	0.505	0.689	0.633	0.626	0.731	0.576	0.568	0.606
ChatGPT OpenAI	0.525	0.905	0.922	0.974	0.964	0.739	0.956	0.876	0.744	0.928	0.907	0.996	0.904	0.740	0.824
FSR															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.653	0.598	0.692	0.649	0.605	0.723	0.735	0.623	0.503	0.709	0.589	0.538	0.709	0.588	0.699
SentiWordNet	0.546	0.542	0.553	0.685	0.685	0.613	0.542	0.611	0.671	0.516	0.673	0.503	0.524	0.760	0.697
VADER	0.560	0.552	0.535	0.511	0.506	0.561	0.547	0.601	0.618	0.567	0.640	0.672	0.581	0.628	0.509
ChatGPT OpenAI	0.914	0.874	0.694	0.972	0.874	0.511	0.974	0.917	0.613	0.968	0.872	0.673	0.887	0.877	0.505
FSR with SMOTE															
Model/Matrix	KNeighbors			SVM			LR			MNB			RF		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
AFINN	0.876	0.737	0.516	0.818	0.817	0.648	0.917	0.915	0.921	0.778	0.773	0.918	0.651	0.651	0.641
SentiWordNet	0.538	0.753	0.529	0.526	0.525	0.917	0.603	0.584	0.648	0.567	0.564	0.604	0.508	0.508	0.514
VADER	0.697	0.734	0.693	0.615	0.612	0.918	0.602	0.595	0.694	0.607	0.603	0.677	0.570	0.563	0.606
ChatGPT OpenAI	0.505	0.745	0.534	0.994	0.954	0.729	0.987	0.917	0.993	0.913	0.729	0.805	0.905	0.999	0.985

TABLE 4.3: ACC, F-measure, and AUC results for the proposed model with cross-validation

Model	ABG			CAP			STD			NED			FSR		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
OpenAI+BERT-BiLSTM (S ²)	0.884	0.869	0.890	0.885	0.858	0.883	0.904	0.897	0.887	0.922	0.878	0.880	0.931	0.928	0.923
OpenAI+BERT-BiLSTM (NSS ³)	0.505	0.497	0.531	0.660	0.676	0.485	0.788	0.716	0.514	0.833	0.831	0.723	0.787	0.787	0.546

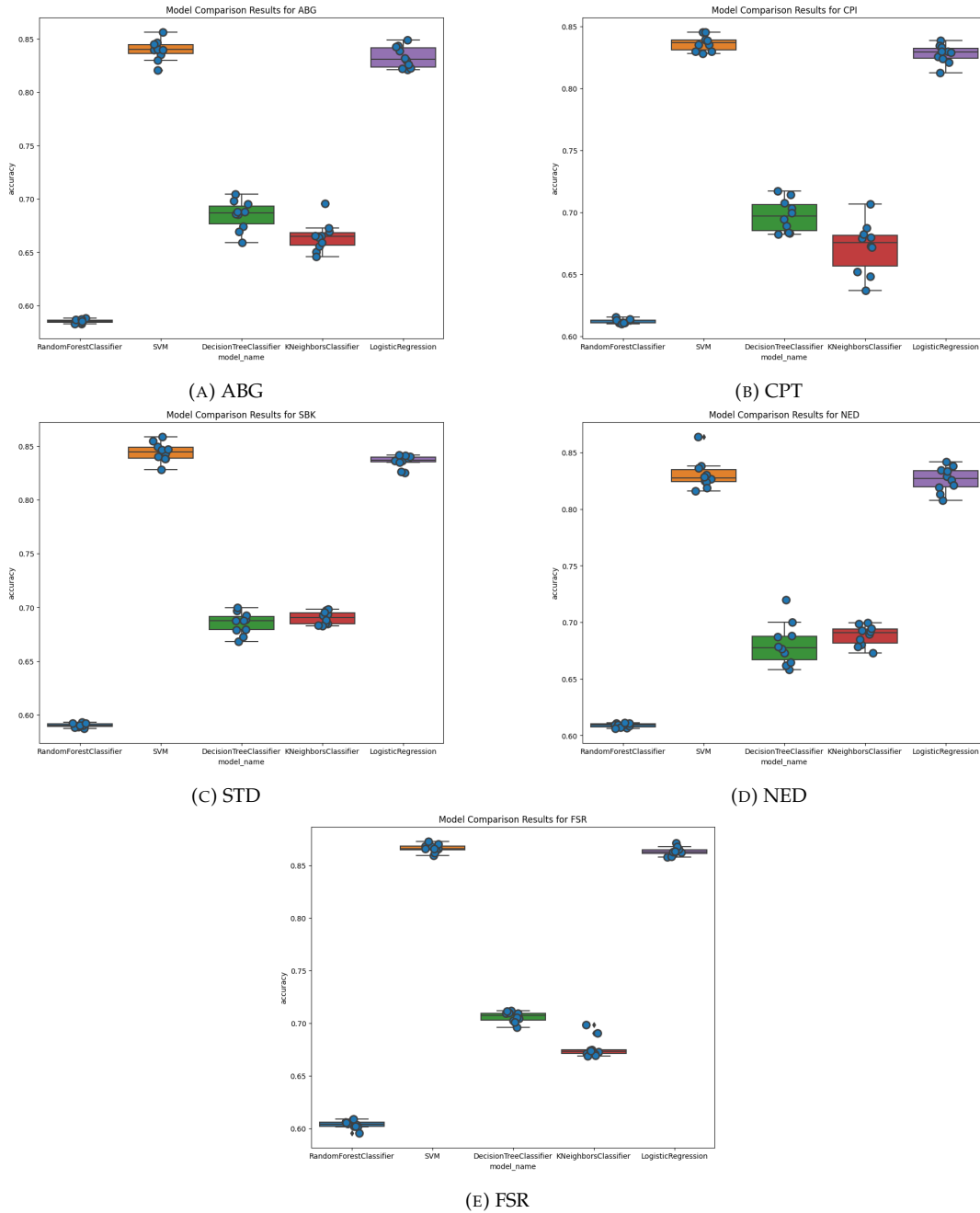


FIGURE 4.7: Performance of different ML models in SA across various datasets.

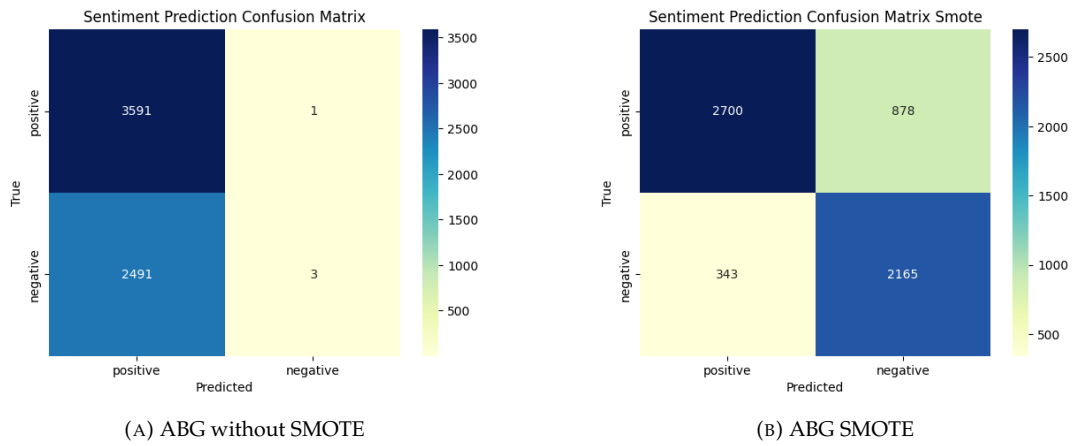


FIGURE 4.8: Confusion matrix performance for ABG dataset.

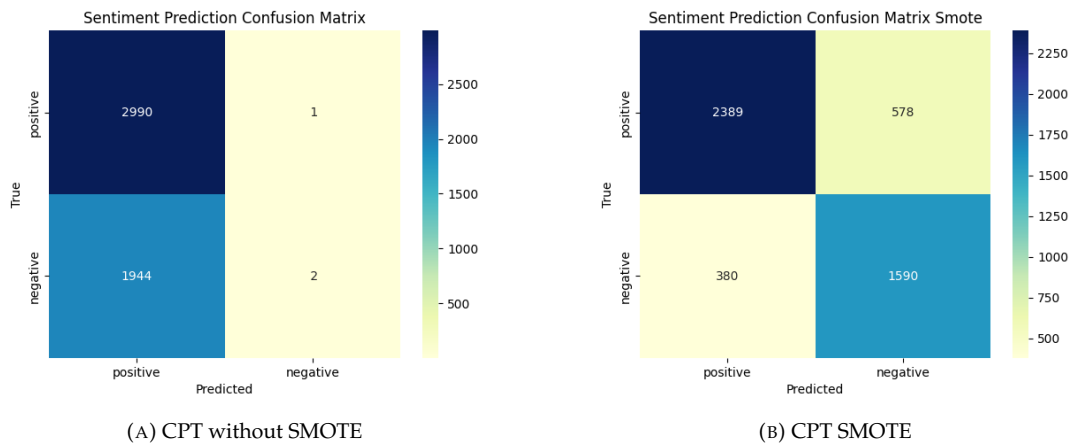


FIGURE 4.9: Confusion matrix performance for CPT dataset.

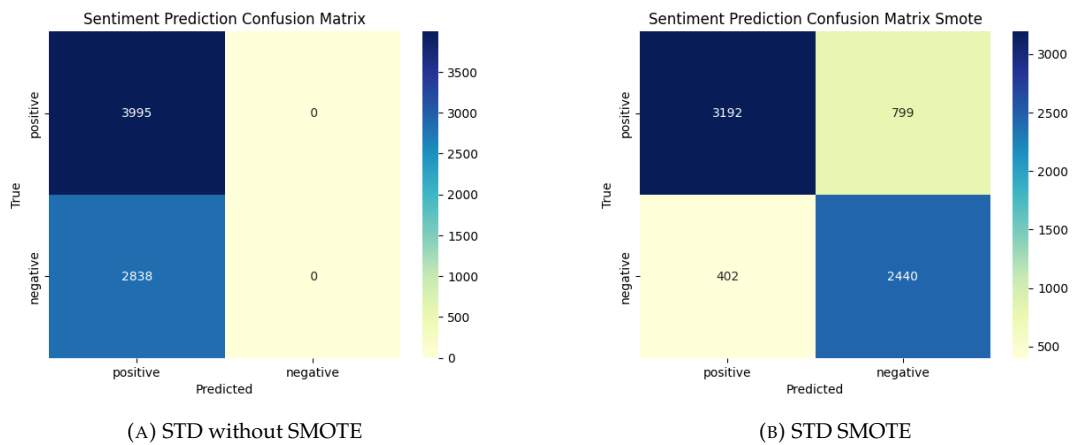


FIGURE 4.10: Confusion matrix performance for STD dataset.

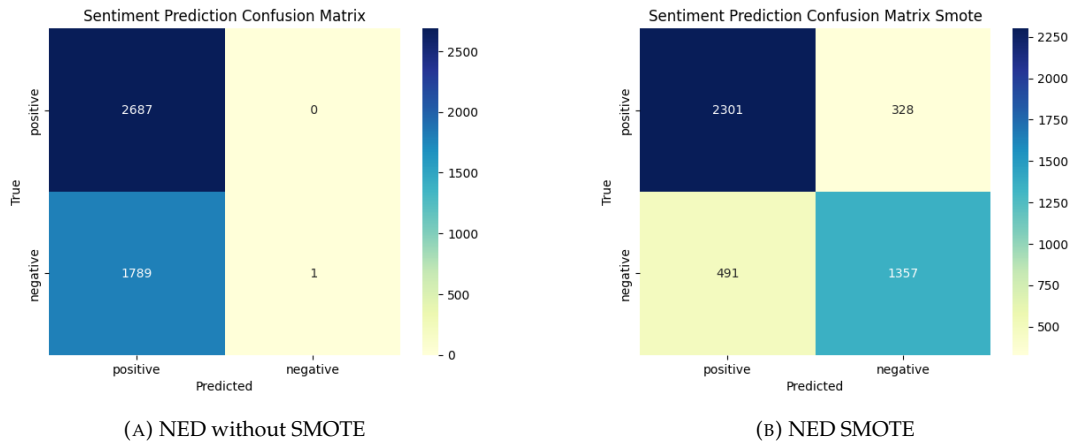


FIGURE 4.11: Confusion matrix performance for NED dataset.

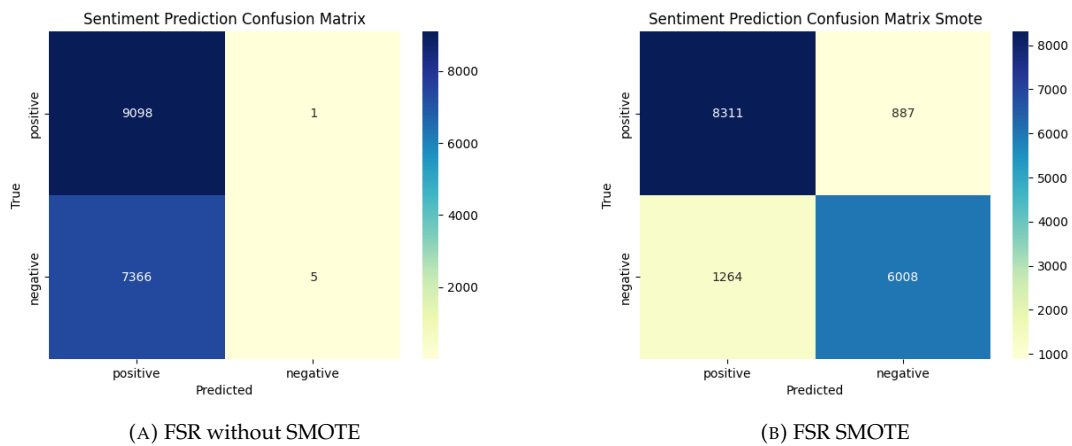


FIGURE 4.12: Confusion matrix performance for FSR dataset.

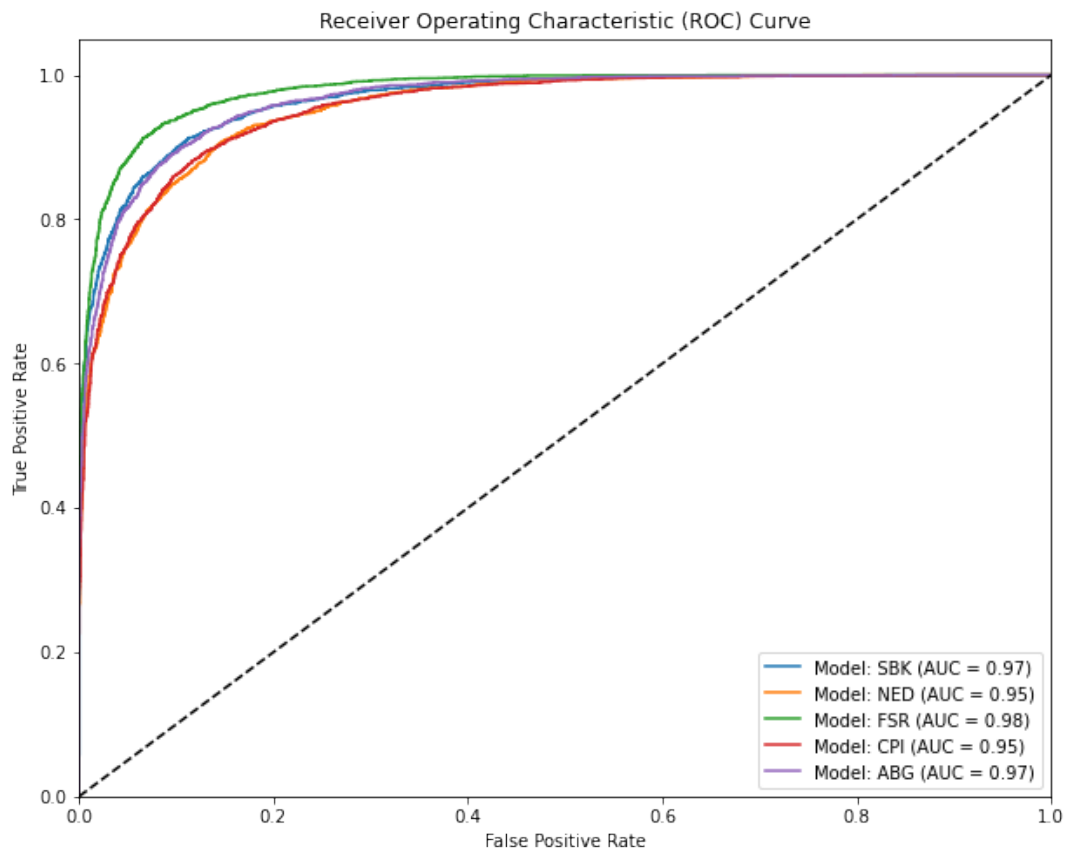


FIGURE 4.13: Area under the ROC Curve with SMOTE techniques on the datasets

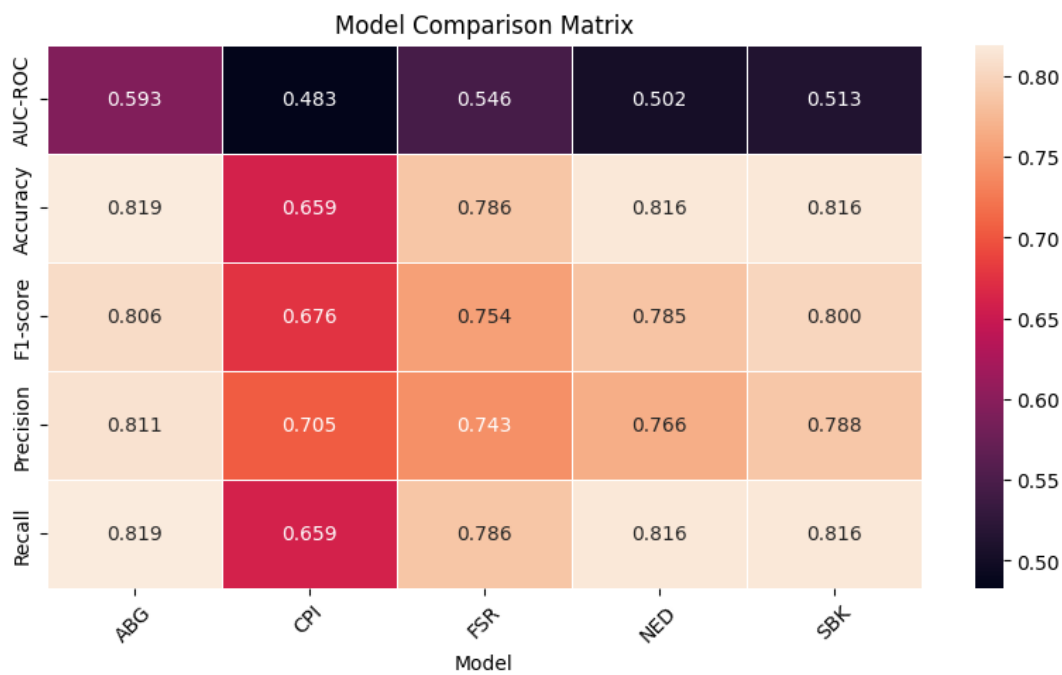
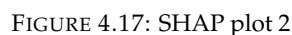
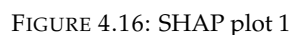
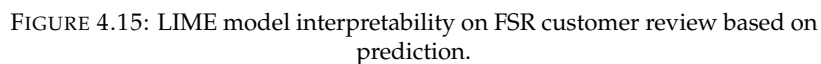


FIGURE 4.14: Proposed model comparison matrix on the datasets



Chapter 5

Retrieval-Augmented Generation with Auto-regressive Language Model for SA

The well-known task of SA has been used to assess customer feedback and media headlines to gauge the personality and potentially polarized sentiment of a given text. Yet with the rise of social media and the online platforms through which we express ourselves—such as X (formerly Twitter), Facebook, Tumblr, and others—the task has taken on an added dimension within the realm of financial institutions. More and more, these firms are apparently using SA to perform customer relationship management by monitoring customer expressions as well as public opinion expressed by customers on their products and services. Supervised learning algorithms have been popularly employed for this task, but the performance of these models has been compromised due to the content’s brevity and the presence of various kinds of business abbreviation terms and financial terminologies in the reviews.

Furthermore, PTLM cannot capture the bidirectional contextual knowledge encoded in word dependencies because it relies on a sentence-level representation that misses the broader picture. To overcome the limitations associated with LBMs and the use of pre-defined sentiment lexicons that may not cover all of the sentiments expressed in natural language or handle the variety of topics and tasks we want to ask questions about, we developed a novel structure called LFEAR. LFEAR combines the RAG model with a conversation format for an ARFT to create an advanced natural language model that serves as a foundation for our work. We then fine-tuned LFEAR on reviews of the HelloPeter, which are infused with the kind of industry-specific contextual information that our model needs to be resilient and flexible in analyzing the sorts of sentiments expressed in a variety of natural language reviews of things like restaurants, movies, politics, and financial products.

The model that we proposed reached an average PR score of 98.45%, a correctness level for answers of 93.85%, and context PR of 97.69% according to the RAGAS metrics. Our LFEAR model conducted effective SA over several different domains and

was equally adaptable and potent in its inference mechanism as it was in conducting SA over the different domains. It was able to take into consideration the unique language features of the different domains and conduct the SA over those domains with the proper annotation of the sentiment and without any bias. These are essentially the benefits of the LFEAR model for social media posts, especially those made by individuals in the financial sector, such as investors and institutions whose posts are largely advantageous to entities listed on the JSE. The LFEAR architecture, proposed as shown in Figure 5.1, aims to make the following contributions.

- We present LFEAR, an advanced inference model that performs complex conversational fine-tuning on auto-regressive LLMs such as Llama-2, Llama-3, GPT-3.5, and GPT-4 to analyze South African financial product reviews.
- The LFEAR combines prompt engineering, RAG, and continuous learning to deal with the features of sarcasm and classify the sentiment in other domains such as financial news, political debate, restaurant reviews, hotel evaluations, and movie reviews.
- We assess the LFEAR using the majority voting method across a broad array of datasets to measure the correlation between public sentiment and institution performance. This suggests that the LFEAR is adaptable and has strategic relevance in many different contexts.
- LFEAR enhances SA by including RAGAS along with sentiment intensity, granularity classification, and vendi score to enhance LLMs performance evaluations. This method addresses the gap in current financial SA metrics, permitting clearer sentiment distinctions and more precise evaluations of customer responses.

To thoroughly verify the proposed LFEAR model, we derive a fundamental mathematical equation that represents the model's essential elements: data ingestion, retrieval augmented generation, fine-tuning, inference, and ensemble voting. This equation unifies the architectural parts of the model, showing how it generates and combines predictions.

Let:

- $\mathcal{D}_{\text{input}}$: Initial input data, including prompts and queries, ingested from both Hellopeter and cross-domain datasets.
- V : The vector database containing embeddings generated from the input data.
- $\text{RAG}(\cdot)$: Function that produces a context-aware representation by retrieving relevant embeddings from V .
- $\mathcal{M}_i$: Each individual model within the ensemble, where $i = 1, 2, \dots, N$ (models such as Llama-2, Llama-3, GPT-3.5 Turbo, etc.).

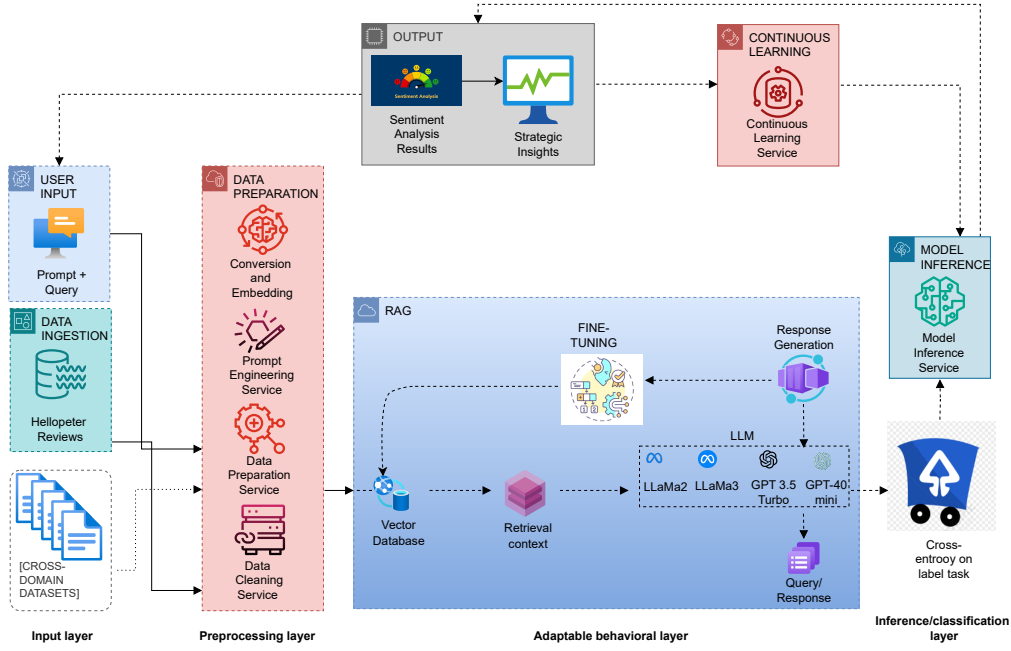


FIGURE 5.1: Proposed LFEAR model for SA.

- w_i : Weight for model $\mathcal{M}_i$, optimized based on its cross-domain performance.
- $f_i(r)$: Sentiment output (positive or negative) from model $\mathcal{M}_i$ for review r .
- $S(r)$: The aggregated sentiment score derived from ensemble predictions.

The overall sentiment prediction for the LFEAR model is then defined as the equation 5.1:

$$S(r) = \arg \max_{s \in \{\text{positive}, \text{negative}\}} \sum_{i=1}^N w_i \cdot \delta(f_i(\text{RAG}(\mathcal{D}_{\text{input}})) = s) \quad (5.1)$$

where:

- $S(r)$ is the final sentiment label for a given review r .
- $\delta(\cdot)$ is an indicator function that returns 1 if the condition inside is true (i.e., if model $\mathcal{M}_i$ predicts sentiment s after being fine-tuned on context from RAG), and 0 otherwise.
- The RAG component $\text{RAG}(\mathcal{D}_{\text{input}})$ serves as the contextualized input for each model in the ensemble, enhancing relevance and ACC.

This equation combines the output from each model in the ensemble, weighted by its relevance and ACC, to produce a robust final sentiment classification $S(r)$.

5.1 Input Layer

The input layer is also the first layer of the proposed architecture, which deals with various inputs from the users and domain datasets. It has a very important function to perform in starting the SA process by incorporating user prompts, queries, and other data sets. The layer effectively integrates real-time customer feedback from other sites, including HelloPeter, with cross-domain data sources such as financial news, tweets of politicians, and restaurant reviews for enhanced analysis. This makes the integration of the multiple modes very effective in making the model very general and effective for use in different SA applications.

5.1.1 Data Ingestion

The data ingestion process brings in user feedback mainly from the HelloPeter platform, where customers in South Africa leave reviews of their financial institutions. These reviews span a number of banks, such as SBK, NED, ABG, CPI, and First National Bank. The model takes in the reviews as data from API endpoints in real-time, meaning the data is in sync with current sentiment trends. This means that the model is working with a near-real-time reflection of what customers think about these institutions. Key aspects of maintaining the model's data quality are discussed in Chapter 3.

5.1.2 Cross-Domain Datasets

We exclusively utilize cross-domain datasets from non-financial sectors, like hospitality and media, for our model's generalizability evaluation. This operation assesses the model's skill at sentiment labeling across different, previously unencountered situations. Orbital checks ensure the model's fitness when applied to diverse and novel contexts, following the rules of domain adaptation and out-of-domain generalization [145]. We cross-test the model on datasets from a range of industries to ensure the model is not overfitting to the finance sector and is instead resilient and adaptable to the type of diverse sentiment expression one is likely to encounter across any industry [146].

5.1.3 User Input

The user input component allows people to perform real-time SA. Users submit natural language prompts or queries to achieve this. We enhance the contextual relevance of these inputs by first classifying them as either positive or negative sentiment. However, some queries consist of conflicting intents or ambiguous sentiment, which prompts the model to use a newly implemented intent conflict detection mechanism to better handle these types of inputs. Instructions that contradict each other and sentiment inconsistencies in the prompts that are flagged by this mechanism make the LLM easier to understand overall [147]. First, our model

uses a default mechanism to determine the conflict type and set the sentiment type. This has worked to make our model more robust in a variety of ways. Consistency in decision-making across different input types is a big step toward a systematic mechanism for achieving accurate sentiment predictions across a range of inputs, as shown in Algorithm 3. Two primary input types are distinguished:

Prompt: A prompt is a pre-defined instruction set that standardizes how the model interprets and classifies sentiment. Defining sentiment polarity, contextual boundaries, and focus areas e.g., customer satisfaction, prompts limit ambiguity and create a framework for consistent, context-aware SA.

Query: A query represents the specific input submitted by the user for sentiment classification. Each query operates within the framework set by the prompt and is enhanced by RAG-derived context, enabling the model to capture subtle sentiment cues. Examples include:

- Query: “Analyse the following customer feedback as positive or negative: The service was fast and good, but the charges are very expensive.”
- Query with Context: “The following are some similar reviews: The staff were helpful but the time taken to serve was long. Please classify whether the review is positive or negative.”

Algorithm 3 User Input Processing for SA with Intent Conflict Detection

Require: User input Q (query) in natural language

Require: Pre-defined prompt P for sentiment classification context

Ensure: Sentiment classification output (positive or negative)

Initialize the prompt P with defined parameters (e.g., polarity, context boundaries, focus areas)

Define $RAG_context \leftarrow$ Retrieve relevant context from previously collected sentiment data

3: Step 1: Query Structuring and Preprocessing

Structure input Q based on the established prompt P for sentiment classification

Clean and normalize Q (e.g., remove unnecessary characters, standardize format)

6: Step 2: Intent Conflict Detection and Resolution

Initialize conflict flag $conflict_detected \leftarrow$ False

if Q contains contradictory instructions or sentiment cues **then**

9: Set $conflict_detected \leftarrow$ True

Resolve Conflict:

 Apply predefined rules for intent conflict resolution or prompt user for clarification

12: **if** clarification received from user **then**

 Update Q based on resolved intent

else

15: Proceed with modified query resolution rules (e.g., prioritize primary sentiment)

Step 3: Sentiment Classification with Contextual Augmentation

 Formulate final input $Q_{final} \leftarrow Q + RAG_context$

18: Apply model to Q_{final} for sentiment classification based on P parameters

 Obtain sentiment output $S \in \{\text{positive, negative}\}$

Step 4: Output Generation and Response

21: Generate final sentiment output S with relevant insights

if $conflict_detected$ **then**

 Append conflict resolution note to output for transparency

24: **return** S

5.2 Preprocessing Layer

The SA process starts with the preprocessing layer, which alters the base data into a format that is more appropriate for embedding and model prediction. This layer contains three major parts: data cleaning, data preparation, and conversion and embedding.

5.2.1 Data Cleaning

Preprocessing the data begins with cleaning it, a necessary step in removing elements that are not needed in the raw text. The aim here is to create a more structured input for the model. The cleaning task itself involves several subtasks to get the text into a more usable form. For HTML data, it involves stripping out the HTML tags. Special characters need to be dealt with, as do stopwords—the words that are so common that they are not useful for making distinctions between one kind of text and another, removing emojis as well to convert the text to lowercase and perform lemmatization. All the cleaning steps need to be done in such a way that the cleaned text preserves the important linguistic structures that the model will use to make its classifications. The function representing this cleaning procedure is shown by equation 5.2:

$$\text{Cleaned Text} = \text{Lemmatise}(\text{RemoveStopwords}(\text{Normalise}(\text{Original Text}))) \quad (5.2)$$

This cleaning function has several steps, and each one takes the raw text and transforms it into a structured format for the model to understand. This allows the model to see the differences between different types of text; they also serve to increase the overall ACC of the model.

5.2.2 Data Preparation

Data preparation is the process that enhances models. It ensures the cleanliness, consistency, and anonymization of the input data. The process of cleaning up the data involves eliminating any unnecessary or potentially harmful elements from the raw data prior to its integration into the algorithm. It also means making the necessary bridged connection across different datasets. A clean, consistent, well-anonymized dataset is also standardization-ready. Additional steps include contraction expansion e.g., “can’t” to “cannot” and accent removal for improved readability. SpaCy is used for tokenization and lemmatization, reducing words to their root forms, which supports consistent analysis and strengthens the reliability of sentiment classification [130, 131]. Through these preprocessing steps, data preparation ensures uniform, relevant input, promoting higher model ACC and robustness.

5.2.3 Prompt Engineering

One of the critical aspects of obtaining relevant sentiment classifications from our model is prompt engineering. In the domain of financial product review SA, we have found that the use of some of the most advanced methods of prompt engineering, like few-shot learning, chain-of-thought reasoning, and ReAct (reason and act),

can dramatically increase our model's capability to understand the nuanced sentiment of the detailed reviews we apply it to. The detailed process of our prompt engineering strategy is outlined in Algorithm 4.

Few-shot Learning [148]: This approach takes advantage of the model's ability to generalize from well-chosen examples that are part of the prompt. Even when the model receives a limited quantity of actual training data, it can typically perform the desired task reasonably well, given that the prompt itself provides sufficient information for the model to determine the nature of the new inputs it must classify [70] as shown in Figure 5.2. One prompt used to assess this ability looked like the following below:

Help me classify this customer review into positive or negative sentiment. Here are some examples:

- "I love the mobile app; it's very user-friendly." (positive)
- "Their customer support is unresponsive and unhelpful." (negative)

Help me classify this customer review into positive or negative sentiment. Here are some examples:

- "I love the mobile app; it's very user-friendly." (positive)
- "Their customer support is unresponsive and unhelpful." (negative)

Review: "I went to open an account it was as fast as I never expected did not even stay 5 minutes on queue I was more than happy and the person who assisted me was a lovely person with good manners and the way of her work is very professional keep it up young lady you will go far"

Review: "I went to open an account it was as fast as I never expected did not even stay 5 minutes on queue I was more than happy and the person who assisted me was a lovely person with good manners and the way of her work is very professional keep it up young lady you will go far"

Sentiment: Positive

FIGURE 5.2: Few-shot Learning with Meta-Llama-3 Model.

Chain-of-Thought Reasoning [149]: The enhancement of the model's interpretability by implementing the logical method it takes to classify sentiment [149], as shown in Figure 5.3. This enables the model to take into account factors like the delivery of services, the characteristics of the product, and the degree to which customers are satisfied prior to making a classification. The prompt that is employed is:

Below is a customer financial product review. Determine if the sentiment is positive or negative. Consider the following aspects: customer service, product features, and overall satisfaction.

Below is a customer financial product review. Determine if the sentiment is positive or negative. Consider the following aspects: customer service, product features, and overall satisfaction.
Review: "I have tried several times to get a IT3b statement for the 201718 tax year I have emailed several times and have not had a response I have registered for online banking but it says that i do not have transactional functionality so i cannot download the statement Can someone please help me"



Review:

"I have tried several times to get a IT3b statement for the 201718 tax year I have emailed several times and have not had a response I have registered for online banking but it says that I do not have transactional functionality so I cannot download the statement Can someone please help me"

Sentiment: Negative

FIGURE 5.3: Chain-of-Thought Reasoning with OpenAI GPT-4o Mini Model.

ReAct (Reason and Act) [150]: ReAct merges reasoning and real-time adaptation in the model. As it considers the query, the model adapts its output and combines reasoning with real-time output adaptation [150]. Figure 5.4 displays this combination and what it looks like in a prompt:

You are a financial product reviewer specializing in South African banks. Your task is to classify customer reviews into positive or negative sentiments. First, reason through aspects such as customer service, product features, and overall satisfaction. Then, act by making your classification.

You are a financial product reviewer specializing in South African banks. Your task is to classify customer reviews into positive or negative sentiments. First, reason through the aspects such as customer service, product features, and overall satisfaction. Then, act by making your classification.
Review: "Still today I am struggling to get assistance with my account Debit order reversals that are not done and they keep charging me"



Review: "Still today I am struggling to get assistance with my account Debit order reversals that are not done and they keep charging me."

Answer: Negative

FIGURE 5.4: Reason and Act with OpenAI GPT-3.5 Turbo Model.

Prompt engineering ensures that the model can capture sentiment and learn the context in which sentiments are expressed, thus enhancing the model's overall SA capabilities. This enhancement provides a particular benefit when examining financial

reviews, which often employ subtle expressions and mixed sentiments.

Algorithm 4 LFEAR Prompt Engineering Techniques

Require: Input text T (customer review in natural language)

Require: Pre-defined prompts for Few-shot Learning, Chain-of-Thought Reasoning, and ReAct techniques

Ensure: Sentiment classification output $S \in \{\text{positive, negative}\}$

- 1: **Step 1: Apply Few-shot Learning**
 - 2: Construct few-shot learning prompt $P_{fewshot}$ with labeled example reviews
 - 3: Generate initial classification prediction $S_{fewshot}$ using $P_{fewshot}$
 - 4: **if** $S_{fewshot}$ indicates ambiguity or mixed sentiment **then**
 - 5: Proceed to Chain-of-Thought Reasoning for further clarification
 - 6: **else**
 - 7: **return** $S_{fewshot}$ as final sentiment classification
 - 8: **Step 2: Apply Chain-of-Thought Reasoning**
 - 9: Construct chain-of-thought prompt P_{CoT} with guiding criteria (e.g., customer service, product features, satisfaction level)
 - 10: Generate refined classification prediction S_{CoT} based on P_{CoT}
 - 11: **if** S_{CoT} is inconclusive or requires real-time adaptation **then**
 - 12: Proceed to ReAct prompt for adaptive reasoning
 - 13: **else**
 - 14: **return** S_{CoT} as final sentiment classification
 - 15: **Step 3: Apply ReAct (Reason and Act)**
 - 16: Construct ReAct prompt P_{ReAct} , focusing on aspects such as reasoning and decision-making
 - 17: Generate final adaptive sentiment prediction S_{ReAct} based on P_{ReAct} and review of T
 - 18: **if** S_{ReAct} is confirmed with high confidence **then**
 - 19: **return** S_{ReAct} as final sentiment classification
 - 20: **else**
 - 21: Flag T for manual review if sentiment classification remains unclear
-

5.2.4 Conversion and Embedding

Once the data is cleaned and processed, it is converted into numerical representations known as embedded. The embeddings used in this thesis are generated by a SentenceTransformer model called *all-MiniLM-L6-v2* [151]. This model is designed to capture the semantic essence of the texts so that their corresponding vectors reflect orders of magnitude similar to those reported as embeddings from word2vec or GloVe. The dimensionality of the vectors for this model is 384, which is also much smaller than the maximum stated for the possible embeddings. The embeddings

are then plugged into a Facebook AI Similarity Search (FAISS) index for efficient searching [152].

The embeddings are derived in the following manner. Here, let $X = \{x_1, x_2, \dots, x_n\}$ be the cleaned and preprocessed set of reviews, where each x_i is a textual input. The embedding model ϕ transforms each x_i into a fixed-size vector $\mathbf{v}_i \in \mathbb{R}^{384}$ as shown by 5.3:

$$\mathbf{v}_i = \phi(x_i), \quad (5.3)$$

where $\mathbf{v}_i$ is the embedding vector for review x_i .

The FAISS index is used to index the embedding vectors. Given a query vector $\mathbf{q}$, the retrieval process finds the k nearest neighbours by minimising the Euclidean distance d between $\mathbf{q}$ and all stored vectors $\mathbf{v}_i$ shown by 5.4:

$$d(\mathbf{q}, \mathbf{v}_i) = \|\mathbf{q} - \mathbf{v}_i\|_2. \quad (5.4)$$

The algorithm returns the k documents with the smallest distance d , indicating to the text with most semantically similar reviews. Embedding and retrieval make SA models scalable and high quality. They work by adding a lot of contextual and semantic information to the input data.

The most relevant contexts for SA are gathered through the generation of embeddings and the retrieval of documents. We use pre-trained embedding models and FAISS [152], both of which are quite effective indexing schemes, to do our document retrieval. The first step in our process is to convert both the query and the top k search results into dense vectors, which are essentially a form of embedding. To do this well—to get the semantic information required for similarity matching—one would ideally use a model that has been specifically pre-trained for the tasks of our interest. After all, the really tricky part of this operation is retrieving the most relevant documents given an input query. Steps for making embeddings and retrieving documents using FAISS are detailed in Algorithm 5.

5.3 Adaptable Behavioral Layer

The model's adaptable behavioral layer is at its center. The model itself combines retrieval-augmented generation with fine-tuning techniques on LLMs, particularly Llama-2 and Llama-3, as well as OpenAI's GPT-3.5 Turbo and GPT-4o Mini. The layer considers the ever-evolving nature of human behavior, allowing the model to continuously learn and situate itself within the specific domains it serves. Especially when handling product reviews concerning South African financial institutions. The model's performance benefits greatly from the RAG component, which significantly

Algorithm 5 Conversion and Embedding Generation

Input: Query q , Document Set $X = \{x_1, x_2, \dots, x_n\}$, Number of neighbours k

Output: Top k most relevant documents D_{top_k}

```

1: Load pre-trained embedding model  $\phi$ 
2: Initialise FAISS index  $I$ 
3: Initialise index-to-document map  $M$ 
4: for each document  $x_i$  in  $X$  do
5:    $v_i \leftarrow \phi(x_i)$  ▷ Generate embedding for document  $x_i$ 
6:   Add  $v_i$  to FAISS index  $I$ 
7:    $M[i] \leftarrow x_i$  ▷ Store document reference for later retrieval
8:  $q_{\text{embedding}} \leftarrow \phi(q)$  ▷ Generate embedding for the query
9:  $[d_{\text{top}_k}, \text{indices}] \leftarrow I.\text{search}(q_{\text{embedding}}, k)$  ▷ Retrieve top  $k$  nearest neighbours
10:  $D_{\text{top}_k} \leftarrow \{M[\text{indices}[j]] \mid j = 1, \dots, k\}$  ▷ Map retrieved indices back to original documents
11: return  $D_{\text{top}_k}$ 

```

enhances the inference process. By weaving in more context and fetching other pertinent information from various resources. LLM performance is improved using fine-tuning techniques, which are used to adapt the LLMs to the financial domain, a highly specialized area that uses not only a distinctive vocabulary but also has its own sentence and structural grammar. These fine-tuned models are now better equipped to do sentiment classification on financial reviews as showing in Algorithm 6

Algorithm 6 Adaptable Behavioural Layer

Require: Query q , Document corpus D , Vector database V , Fine-tuned language models $\{\text{Llama-2, Llama-3, GPT-3.5 Turbo, GPT-4o Mini}\}$

Ensure: Final sentiment classification output S

1: **Step 1: Query Embedding Creation**

2: Encode the input query q into an embedding $\mathbf{q}$ using the pre-trained embedding model (e.g., 'all-MiniLM-L6-v2')

3: **Step 2: Vector Database Retrieval (FAISS)**

4: Use FAISS to retrieve the top- k relevant document embeddings $\{d_1, d_2, \dots, d_k\}$ from V based on the Euclidean distance between $\mathbf{q}$ and each document embedding $\mathbf{v}_i$ in V shown below by equation 5.5:

$$d(\mathbf{q}, \mathbf{v}_i) = \|\mathbf{q} - \mathbf{v}_i\|_2 \quad (5.5)$$

5: **Step 3: RAG**

6: **for** each retrieved document d_i in $\{d_1, d_2, \dots, d_k\}$ **do**

7: Concatenate q with d_i to form a combined input

8: Pass the combined input to each fine-tuned language model to generate an intermediate response r_i with probability shown below by 5.6:

$$P(r | q) = \sum_{d \in D_k} P(r | q, d) P(d | q) \quad (5.6)$$

9: **Step 4: Fine-Tuning with Domain-Specific Data**

10: **for** each model M in $\{\text{Llama-2, Llama-3, GPT-3.5 Turbo, GPT-4o Mini}\}$ **do**

11: Fine-tune M on financial review data, using techniques such as LoRA and 4-bit quantization for efficient adaptation

12: Update model parameters with optimized hyperparameters (learning rate, batch size, gradient accumulation) as needed

13: **Step 5: Sentiment Response Generation**

14: **for** each model M in $\{\text{Llama-2, Llama-3, GPT-3.5 Turbo, GPT-4o Mini}\}$ **do**

15: Generate a sentiment classification output S_i for the query q using model M

16: **Step 6: Ensemble Voting for Final Output**

17: Combine the outputs $\{S_1, S_2, S_3, S_4\}$ from each model using majority voting to determine the final sentiment classification S

18: **return** S

5.3.1 Model Selection

The reason for selecting certain models revolved around their ability to process language, retain context, and adapt to domain-specific scenarios. Among the chosen models, Llama-2 and Llama-3 stand out for their efficiency in managing the complex and emergent language patterns of reviews for financial products [153]. The team

also selected GPT-3.5 Turbo and its successor, GPT-4o Mini, because they are known to generate contextually aware responses, even when the language being processed is complex [154]. Each model was chosen based on its architecture and pre-training corpus, which together make a strong base for much more adaptation through fine-tuning [155]. Llama-2 and Llama-3, for instance, handle not just ordinary language but also the intricate structures that certain kinds of specialized, domain-specific language take with seeming ease [156]. Their proficiency makes them ideal for tasks where a simple understanding of ordinary sentiment and context won't suffice, as the language in question is not at all simple. Indeed, any model we employ must proficiently handle both ordinary tasks and unconventional ones. The fine-tuning of each model is discussed in detail in Subsection 5.3.5.

5.3.2 Vector Database Implementation

The implementation of the vector database relies on FAISS [152] for searching similar vectors in what could be thought of as a high-dimensional neighborhood—each point in this neighborhood being a vector representing some object. The objects we're concerned with in this work are words, which are represented by vectors that we generate using the fast and efficient transformer model known as 'all-MiniLM-L6-v2' [151]. These word vectors are stored in the vector index we maintain—a highly efficient and effective indexing mechanism that allows the model to search for similar words, or in our case, retrieve similar words with high ACC and low latency.

In practical application, employing a single vector database for context retrieval in a conversational agent can potentially impact its scalability. Because as the amount and variety of data grow, so does the load on any single database. We could address this problem in the future by architecting the application in such a way that it uses either distributed or hierarchical indexing techniques. These methods, basically, allow us to subdivide our data retrieval process into a number of more partitions—working our way down through several layers of indices to find just the right piece—a process involving several orders of magnitude more search space that is just as efficient and a lot more accurate than if we were searching through a single database.

5.3.3 Retrieval-Augmented Generation

The RAG architecture merges document retrieval with generative SA, thus enhancing the context relevance of the output produced by the model. Within this setup, relevant documents are first pinpointed from a vector database and then provided to the LLMs for the specific task of sentiment classification. This is particularly useful for tasks that are specific to certain domains, such as the financial sector, as illustrated in Algorithm 7.

The process of retrieving information starts with both the query and the document corpus being encoded into dense embeddings via a pre-trained embedding model (e.g., all-MiniLM-L6-v2). FAISS is utilized next for vector database search to efficiently find the top k relevant documents corresponding to each query. For each dataset, the value of k is chosen to give the balance between retrieval speed and contextual relevance. Based on the results of our validation tests, we have found that $k = 5$ seems to provide this ideal balance. In addition, cosine similarity has been selected as the distance metric within the FAISS index to optimize matching PR.

The generative ability of LLMs is enhanced when they are used in conjunction with an information retrieval system. The RAG system can be thought of as having two parts: the retriever and the generator. The retriever uses FAISS, a library for fast nearest neighbor search in high-dimensional spaces, to locate relevant documents among potentially billions of them. The generator takes those documents and conditions generation on them, meaning that the good old probabilistic language modeling that happens in an LLM occurs while also keeping in mind the newly retrieved material. We configured the LLMs' decoding settings with a beam width of 3, a temperature of 0.7, and top-k sampling of 40.

In RAG, the probability of generating a response r given a query q is defined as 5.7:

$$P(r | q) = \sum_{d \in D_k} P(r | q, d)P(d | q) \quad (5.7)$$

This equation sums over the top- k retrieved documents D_k , where $P(r | q, d)$ represents the probability of generating the response given the query and a specific document, and $P(d | q)$ denotes the probability of retrieving the document given the query. The model becomes more proficient at managing intricate or specialized data settings by using a weighted method to serve up responses. This RAG model takes the next step by combining retrieval and generation into a single, contextually sensitive task, allowing it to produce more relevant and enlightened output [157]. RAG is well-suited for pure sentiment-analysis tasks, which often rely heavily on context and necessitate picking the relevant word out of a dense space of specialized knowledge.

5.3.4 Querying the Vector Database

The system performs similarity-based searching using FAISS to obtain the top k documents associated with a specific query, such as customer complaints and comments about banking services. It starts with embedding creation, where new customer service reviews are encoded into embeddings using a pre-trained embedding model (e.g., 'all-MiniLM-L6-v2'). This model is good at taking the semantic information present in the reviews and transforming it into dense vectors [158]. These embeddings of the new banking service reviews are then added to the FAISS index. Once

Algorithm 7 RAG with Specific Retrieval Methodology and Parameter Tuning

Input: Query q , Document corpus D , Pre-trained language model ϕ

Output: Generated response r

- 1: Encode query q and documents in D into embeddings using all-MiniLM-L6-v2
 - 2: **Retrieval Methodology:** Retrieve top- k relevant documents D_k from FAISS index based on cosine similarity; D is indexed with HNSW (Hierarchical Navigable Small World) to balance speed and ACC for high-dimensional vectors
 - 3: **Parameter Tuning:** Set $k = 5$ for retrieval depth, cosine similarity for distance metric, LLM decoding parameters: beam width = 3, temperature = 0.7, top- k sampling = 40
 - 4: **for** each document d_i in D_k **do**
 - 5: Concatenate query q with d_i to create context-enhanced input
 - 6: Generate response r_i using language model ϕ with parameter-tuned decoding
 - 7: Aggregate responses r_i by majority voting to form the final response r
 - 8: **return** r
-

they are in the index, the reviews can be queried with proper semantic understanding, and you can get back suitable banking service complaints as the top results [159].

A global dictionary holds all the reviews. Each entry is linked to its index position and contains the actual review content, making it simple to uncover any original review when the need arises. To find responses, a query is first encoded as an embedding using the same pre-trained model as before. Then the nearest neighbors are found using the FAISS index [160]. With the most relevant documents in hand, the RAG system generates human-like responses to the encoded queries it has been fed. The Euclidean distance measures the similarity between the query embedding $\mathbf{q}$ and each document embedding $\mathbf{v}_i$, as shown below by equation 5.8:

$$d(\mathbf{q}, \mathbf{v}_i) = \|\mathbf{q} - \mathbf{v}_i\|_2 \quad (5.8)$$

5.3.5 Models Fine-Tuning

The fine-tuning of LLMs such as Llama-2, Llama-3, GPT-4o Mini, and GPT-3.5 Turbo was done with the goal of tailoring these models to the special language and cultural peculiarities of customer evaluations in the South African financial industry. This method used domain-specific data, targeted prompts, and sophisticated techniques such as low-rank adaptation (LoRA) [161] and quantization to improve model performance in SA tasks. The Llama-2 and Llama-3 models (e.g., meta-llama/Llama-2-7b-hf and meta-llama/Meta-Llama-3-8B-Instruct) were fine-tuned using a dataset of annotated customer reviews, reflecting sentiment expressions particular to the South African environment. The fine-tuning was led by a cross-entropy loss function that was intended to reduce classification mistakes. The technique employed involved utilizing already established models with well-organized input-output pairs during

the training stage. The prompts used were structured according to the Few-shot Learning, Chain-of-Thought reasoning, and ReAct paradigms. The models' outputs were then aligned with the designated tasks. The task at hand was to get the models to understand the finely detailed sentiment expressions in the financial review domain so the models could capture and thus understand the financial terminologies. Training the models to understand sentiment expressions in financial reviews, culture-relevant expressions, and financial abbreviations [11].

To enhance computing efficiency, we reduced the need for resources by using 4-bit quantization and improved the model performance in doing so [162]. We easily incorporated LoRA [163] into this setup to allow for fine-tuning with much fewer resources, using just a subset of weights to accomplish the task with a level of flexibility that makes such fine-tuning almost universally doable in resource-limited settings. The setup for fine-tuning utilized a learning rate of 1×10^{-4} , a batch size of 4, and an 8-step gradient accumulation. To maintain a balance between model validation and ensuring a nimble response to the inputs it receives, we employed a cosine schedule for the learning rate. The OpenAI models, *gpt-4o-mini-2024-07-18* and *gpt-3.5-turbo-0125*, were fine-tuned in parallel using an identical technique and benefitted from the extra flexibility offered by OpenAI's fine-tuning API. The Helloworld dataset for South African financial reviews was used to train the models, and the prompts used to engage the models were exceptional in their clarity and specificity. While we fully expect the models to have an understanding of English in any context, these measures were taken to ensure that a fine-grained interpretation of the regional idioms, customer expectations, and the necessary sentiment subtleties specific to this local market were achieved.

The dynamic hyperparameter tuning method was used to modify the hyperparameters, including number of epochs, batch size, and learning rate, that the model required to fit the training data well. These were not simply set to fixed values; rather, they were chosen dynamically based on different aspects of the dataset [164]. Where the model was allowed to fit too well to the training data, it did not perform well in terms of generalization to the kinds of contexts represented in the review being evaluated with traditional metrics like ACC, PR, RC, and F-Measure. The Llama and GPT models both achieved substantial and commendable increases in the ACC of sentiment classification after the fine-tuning process. This outcome emphasizes the efficacy of these models for performing SA accurately in a particular domain. The use of fine-tuned conversing models highlights the importance of tailoring LLMs to perform well in specific industrial contexts, especially when the ways in which customers express their opinions are influenced by regional or cultural factors. The personalized approach of our models allows them to handle the complexity of finance better, making them more useful as tools for strategic decision-making in the study of financial sentiment.

5.3.6 Response Generation

The gathered and stored documents provide the context necessary for the Llama and GPT models to perform nuanced SA that is both richly contextual and domain-specific. Every individual model produces a sentiment classification “positive” or “negative” that pertains to the query and the relevant retrieved documents. For any sort of ambiguous or nuanced sentiment where the models may not reach a strong consensus, we get around this problem by employing a confidence score that is based on how well the models agree. A confidence score for the final sentiment prediction S_{final} is calculated based on the proportion of models that classified the sentiment similarly [165, 166]. This score evaluates the reliability of sentiment classification and evaluate the confidence of our models. If we receive low confidence, we promptly implement further verification to ensure ACC. Here’s how we express the confidence score using equation 5.9:

$$\text{Confidence}(S_{\text{final}}) = \frac{1}{n} \sum_{i=1}^n \delta(S_i, S_{\text{final}}) \quad (5.9)$$

where: $-S_{\text{final}}$ represents the majority sentiment classification (positive or negative), $-S_i$ is the sentiment prediction from the i -th model, $-n$ is the total number of models, $-\delta(S_i, S_{\text{final}})$ is a function that returns 1 if $S_i = S_{\text{final}}$ and 0 if $S_i \neq S_{\text{final}}$.

5.4 Inference Layer

The inference layer facilitates the deployment of the trained models for classification and regression tasks. It integrates seamlessly with the overall system, allowing for real-time SA and decision-making in a production environment.

We have constructed and enhanced the model’s architecture for English financial reviews. This could constrain the model’s ability for SA and classification in low-resource South African languages. Possible solutions include the future use of multilingual embeddings and tuning on corpora in our national languages by use of transfer learning techniques. All these methods could help make the model more broadly useful and indeed robust for understanding the kinds of sentiment expressed in financial reviews in low-resource languages

5.4.1 Cross-Entropy on Label Task

The system fine-tunes the LLMs with the help of sentiment-labeled data and a primarily cross-entropy loss function. The cross-entropy function is a standard loss function for comparing how similar or different two probability distributions are. In our case, one of the distributions comes from our model, and the other is the set of correct answers (ground truths) that the model should have predicted instead of what it did predict. The function is defined as equation 5.10:

$$\text{CrossEntropy}(y, \hat{y}) = - \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (5.10)$$

where y_i represents the ground truth label for the i -th instance, $\hat{y}_i$ is the predicted probability for the i -th instance, and n is the total number of instances. This method guarantees that the model increases its classification ACC over time by minimizing the discrepancies between predicted and true outcomes [167].

5.4.2 Model Inference

Sentiment classification is performed using an assembly of Llama-2, Llama-3, and GPT models that have been fine-tuned for the task. The models generate sentiment predictions from the context supplied by the RAG system, and the final sentiment label is adjudicated using a majority voting mechanism. The ensemble's final decision is thereby rendered more robust and less prone to the kind of individual model bias that can happen when context isn't clear [168] as shown below by equation 5.11:

$$r_{\text{final}} = \arg \max_{r \in R} \sum_{i=1}^n \delta(r_i, r) \quad (5.11)$$

where:

- r_{final} is the final sentiment label.
- R represents the set of all possible sentiment responses (e.g., positive or negative).
- r_i is the sentiment prediction from the i -th model.
- n is the total number of models.
- $\delta(r_i, r)$ is an indicator function, defined as 5.12

$$\delta(r_i, r) = \begin{cases} 1, & \text{if } r_i = r \\ 0, & \text{otherwise} \end{cases} \quad (5.12)$$

The function of indicator, $\delta(r_i, r)$, gives the value 1 when the model prediction r_i matches the target response r and gives the value 0 otherwise. The equation is used to calculate r_{final} , which is the sentiment label with the highest number of model votes. This voting mechanism ensures that the sentiment classification is consistent across different model outputs [169].

5.4.3 Continuous Learning

The model supports continuous learning by operating with datasets from a diverse range of global domains, encompassing far more than simply South African customer evaluations. Consequently, the model now comprehends and is considerably less biased in comprehending the full array of ways in which a much wider variety of people express their sentiments and opinions; thus, it is significantly more human-sentiment literate. The model is also vastly more scalable, regardless of the kinds of in-country segmentation that tend to tether AI initiatives. Finally, when the model is placed in a scenario where it must immediately incorporate a new batch of intelligence, it can do so without losing any of the gains it has made in understanding human sentiment.

Continuous learning parameters θ are updated over time as shown by equation below 5.13

$$\theta_{t+1} = \theta_t + \eta \cdot \nabla_{\theta} \mathcal{L}(\theta_t; \mathcal{D}_{\text{new}}) \quad (5.13)$$

where: - θ_t are the model parameters at time t , - η is the learning rate, - $\nabla_{\theta} \mathcal{L}(\theta_t; \mathcal{D}_{\text{new}})$ is the gradient computed over new cross-domain data $\mathcal{D}_{\text{new}}$.

The vector database is also regularly updated by adding new embeddings as shown by 5.14:

$$V_{t+1} = V_t \cup \{\mathbf{v}_{\text{new}}\} \quad (5.14)$$

where: - V_t is the vector database at time t , - $\mathbf{v}_{\text{new}}$ is the embedding of recent data.

5.4.4 Output

The final sentiment classification occurs through a simple majority voting process. Our ensemble of fine-tuned models—Llama-2, Llama-3, GPT-4o Mini, and GPT-3.5 Turbo—produce output classifications that are all together for the final decision. This ensemble method increases the reliability of our decision because it creates a balance between what we think are individual model biases associated with each model's output. And we think the end result is a more even, more reliable path to the final sentiment classification.

The sentiment score S for each review r is calculated as shown by 5.15

$$S = \frac{1}{N} \sum_{i=1}^N w_i \cdot f_i(r) \quad (5.15)$$

where: - N is the total number of models, - w_i is the weight assigned to model i based on its cross-domain performance, - $f_i(r)$ is the sentiment prediction from model i for review r .

To ensure cross-domain consistency, model weights w_i are optimized by minimizing the domain-adaptation loss as shown by 5.16:

$$\mathcal{L}_{\text{adapt}} = \frac{1}{M} \sum_{j=1}^M (S_{j,\text{HelloPETER}} - S_{j,\text{cross-domain}})^2 \quad (5.16)$$

where: - M is the number of test reviews across both HelloPETER and cross-domain data, - $S_{j,\text{HelloPETER}}$ and $S_{j,\text{cross-domain}}$ represent the sentiment scores for review j in each dataset.

The final sentiment label S_{final} is determined by majority voting as shown by equation 5.17:

$$S_{\text{final}} = \arg \max_{s \in \{\text{positive}, \text{negative}\}} \sum_{i=1}^N \delta(f_i(r), s) \quad (5.17)$$

where: - $\delta(f_i(r), s) = 1$ if $f_i(r) = s$ (indicating that model i predicts sentiment s), and 0 otherwise.

5.5 Experiments

The section presents the results from the SA model and provides a detailed account of the model's performance, adaptability to various domains, and some insights obtained from the classification tasks. The model is evaluated using several key metrics: ACC, PR, RC, F-Measure, and RAGAS, which serve to demonstrate the effectiveness of the model and training process with integrating diverse datasets, performing continuous learning, and using RAG techniques. We look closely at how fine-tuning the model on data from specific domains affects the model and makes it work better in those domains. Next, we examine the model's innate robustness and how well it scales and generalizes across various domains. We then delve into RAGAS, which is used to evaluate how well the model performs in terms of the relevance, ACC, and coherence of its generated responses.

5.5.1 Data Acquisition

The main data for this study were obtained from HelloPETER, a well-known South African review platform for financial service providers. Using the API endpoints

Algorithm 8 Fetching and Processing Hellowater Reviews

Input: List of API endpoints (URLs), Maximum pages to retrieve ($P_{\max}$)

Output: Collection of processed reviews $R_{\text{processed}}$

```

1: Initialize  $R_{\text{processed}} \leftarrow \{\}$ 
2: for each  $url$  in URLs do
3:    $response \leftarrow$  Fetch initial data from  $url$ 
4:    $P_{\text{total}} \leftarrow \min(\text{Extract total pages}, P_{\max})$ 
5:   for  $p = 1$  to  $P_{\text{total}}$  do
6:      $data \leftarrow$  Fetch data from page  $p$ 
7:     if data is valid then
8:       Extract relevant fields:  $created\_at$ ,  $review\_title$ ,  $review\_content$ , etc.
9:        $review \leftarrow$  Combine relevant fields into a single text string
10:       $R_{\text{processed}} \leftarrow R_{\text{processed}} \cup \{review\}$ 
11: return  $R_{\text{processed}}$ 

```

corresponding to several major banks such as SBK, NED, ABG, CPI, and First National Bank as shown in Table 5.1. The datasets from different domains were included in the model to diversify contexts and sentiment expressions, thereby augmenting SA. These datasets' cross-domain nature serves to strengthen the model with varied sentiments in even more varied situations. A summary of the datasets appears in Table 5.2.

The varied datasets guarantee that the model stays flexible in different sectors, with PR and ACC ensured. We devised a data extraction algorithm detailed in Algorithm 8 to retrieve and process the Hellowater reviews efficiently. The review data is taken from the API endpoints by the algorithm and moved into a coherent structure for it to be analyzed. The reviews are extracted from several pages across the web, and during this page scripting, certain precautions are taken so that irrelevant and worthless data don't get mixed into the final batch of reviews.

TABLE 5.1: Hellowater Dataset Summary

Dataset	Train	Validation	Test	Total Size	Tokens
Hellowater Reviews	103,472	12,934	12,935	129,341	8,340,421

TABLE 5.2: Cross-Domain Experimental Datasets Description

Datasets	Domain	Total Size	Tokens
SemEval-2014 [120]	Restaurants	2,503	54,086
IMDB-50k [121]	Movies	50,000	13,974,161
TripAdvisor [122]	Hotels	20,491	2,389,961
SemEval-2016 Stance [123]	Politics	26,750	604,358
Financial PhraseBank [124]	Financial News	2,264	50,891

5.5.2 Experiment Setup

We took four top-tier language models and performed our own fine-tuning on them: GPT-4o-mini (gpt-4o-mini-2024-07-18), GPT-3.5 Turbo (gpt-3.5-turbo-0125), Llama2 (meta-llama/Llama-2-7b-hf), and Llama3 (meta-llama/Meta-Llama-3-8B-Instruct). We used the models with a specific SA task relevant to the South African financial sector. The GPT-4o-mini model, which has a context window of 128,000 tokens and a maximum output of 16,384 tokens, presented an economical and efficient solution. For comparison, we used GPT-3.5 Turbo, which has a window size of 16,385 tokens. We also used the Llama models, both Llama2 and Llama3. These are robust models that are good at doing long context-insensitive review tasks.

We performed all experiments on a Google Cloud Platform (GCP) instance equipped with two NVIDIA A100 GPUs (40GB each) and an Intel Cascade Lake CPU. The setup was running a DL VM with CUDA 11.8 and Python 3.10. We fine-tuned the models with a per-device batch size of 4, gradient accumulation steps of 8, and a learning rate of 1×10^{-4} —all while using mixed PR for efficient computation (fp16). We fine-tuned the model over 5 epochs, evaluating it every 20% of the way through the training process. This ensured that the optimal model was found—one that was not just adapted to our specific type of sentiment but also capable of generalization across various types of sentiment expressions. Table 5.3 provides a summary of the hyperparameters and training settings for each model.

TABLE 5.3: Summary of Fine-Tuned Models and Hyper-parameters

Model	Context Window	Max Output Tokens	Training Data	Trained Tokens	Learning Rate	Batch Size	Epochs
GPT-4o-mini-2024-07-18	128,000 tokens	16,384 tokens	Up to Oct 2023	3,909,592	1.8×10^{-4}	12	3
GPT-3.5-turbo-0125	16,385 tokens	4,096 tokens	Up to Sep 2021	4,617,049	2×10^{-4}	10	3
Llama-2-7b-hf	8,192 tokens	4,096 tokens	Up to Dec 2022	-	1×10^{-4}	4	5
Llama-3-8B-Instruct	8,192 tokens	4,096 tokens	Up to Dec 2023	-	1×10^{-4}	4	5

5.5.3 Data Augmentation

SA often runs into a data imbalance problem, particularly in customer feedback contexts, where, for whatever reason, negative sentiments seem to be massively over-represented compared to positive sentiments. In our research, to minimize this well-documented tendency and hopefully prevent our model from internalizing it, we used a variety of methods during the training phase that are designed to make the training more fair and the samples more diverse. This is particularly important because our dataset contained a tremendous amount of negative sentiment with 70% of the reviews in it were classified as negative.

The main method used was to dynamically adjust the loss function so that higher penalties were applied when the model classified the minority of the classes as positive sentiment when they were not [170]. This method kept the model sensitive to underrepresented classes and allowed it to learn in a more balanced way across

the sentiment categories. We also used stratified sampling. This maintains the distribution of classes when creating mini-batches, ensuring that the even mixture of positive and negative examples is presented to the model during training [171]. We lowered model bias and improved our model's generalization capability to produce more equitable and accurate sentiment classifications by applying these methods. Their applicability ranges even to high-stakes areas like financial SA, where a biased model could produce predictions that lead to suboptimal decisions.

5.6 Performance Measurement

We evaluate the SA model using RAGAS, scoring the intensity of sentiment and classifying sentiment at a more granular level. The Vendi Score is used together with traditional evaluation metrics like F-measures and ACC.

5.6.1 Evaluation Metrics

We used traditional metrics to evaluate the performance of our SA models. These include ACC, PR, RC, and the F-measure. Together, these metrics furnish a thorough assessment of how well our models predict sentiment in the positive and negative classes. This section is discussed in detail in Subsection 4.4 of Chapter 4

5.6.2 Retrieval-Augmented Generation Assessment

We include distinct RAGAS metrics to gauge how well our RAG models function. More precisely, we apply metrics that evaluate the quality of the answers our models produce: answer relevance, answer correctness, answer similarity, answer recall, and answer precision [88]. The relevance metric checks to see if the generated answer by our model is close to something relevant, which, in human terms, means it's at least conversationally sensible. To gauge relevance, we use cosine similarity on the embeddings of our model's answer and the original question as shown by equation below 5.18:

$$\text{Answer Relevance} = \frac{1}{N} \sum_{i=1}^N \cos(\mathbf{E}_{a_i}, \mathbf{E}_o) \quad (5.18)$$

where:

- N is the number of generated answers.
- $\mathbf{E}_{a_i}$ is the embedding of the generated answer.
- $\mathbf{E}_o$ is the embedding of the original question.

The answer correctness metric measures how accurate the generated answer is, taking into account both semantic similarity and factual similarity. To arrive at the correctness score, we use the following method as shown by equation below 5.19:

$$\text{Answer Correctness} = \alpha \times \text{Semantic Similarity} + (1 - \alpha) \cdot \text{Factual Similarity} \quad (5.19)$$

where:

- α is the weight assigned to semantic similarity.
- **Semantic Similarity** is the cosine similarity between the embeddings of the generated answer and the ground truth.
- **Factual Similarity** measures the overlap of factual information between the generated answer and the ground truth.

We can measure how well our RAG system works using these specific metrics for RAGAS. They let us look closely at aspects of what the RAG system produces. With them, we can make a kind of experiment to see how well it does at generating answer correctness, answer relevance, and answer similarity. We can also look at how the model performs in answering questions with different types of sentiment polarity.

5.6.3 Integrative Sentiment Analysis Metrics

We use three essential measures in our SA approach to ensure a deep and precise understanding of our customers' feedback: the intensity of the expressed sentiment, the granular classification of the sentiment, and the Vendi Score [172]. We use VADER to calculate the intensity of the expressed sentiment. It assigns one of four levels to each piece of feedback, from strongly positive to strongly negative, capturing both the polarity and the strength of the expressed sentiment. A second key measure in our analysis is the classification level of the sentiment expressed. This is important because, unless one is using a very simplistic model, sentiment expressed is rarely simply positive or negative. We aim to classify expressed sentiment at a level where one can only classify it wrongly if one isn't being at least somewhat nuanced in terms of what different types of sentiments might mean. Finally, the Vendi Score provides an overall ACC metric, representing the ratio of correct predictions to total samples, which helps assess the model's reliability across all sentiment levels clearly and understandably [173]. Measuring Diversity with the Vendi Score is expressed in the mathematical equation below 5.20:

$$\text{VS}_k(x_1, \dots, x_n) = \exp \left(- \sum_{i=1}^n \lambda_i \log \lambda_i \right) \quad (5.20)$$

where:

- λ_i represents the i -th eigenvalue of the kernel matrix K/n ,
- K is a similarity matrix where each entry $K_{i,j} = k(x_i, x_j)$ is defined by a similarity function $k : X \times X \rightarrow \mathbb{R}$.

5.6.4 Performance Evaluation

Table 5.4 provides an overview of the performance of our fine-tuned models on the Hellopeter dataset, which was our main dataset for fine-tuning. All models performed well, showing that they could handle sentiment classification in a finance context. Our best model, *gpt-4o-mini-2024-07-18*, achieved an ACC of 89.15%, closely followed by *gpt-3.5-turbo-0125* with 88.95%. Our next model, *Llama-2-7b-hf*, was a little lower at 81.15%, but still showed some impressive numbers: 88.05% for PR and 88.10% for the F-Measure. The last model we looked at, *Meta-Llama-3-8B-Instruct*, wasn't far behind at 87.05% ACC. So both Llama models were clearly very competitive, and all the models showed they could write South African financial sentiment classifications.

The confusion matrices for the Llama models, shown in Figures 5.5 and 5.6, display the outcomes on the Hellopeter dataset. They reveal a classification that is almost perfectly balanced between the two sentiment classes, as reflected in our evenly distributed results. This perfect balance is indicative of a low misclassification rate that, in turn, is associated with high RC and PR. In Figures 5.7 and 5.8, the confusion matrices of the GPT models are displayed, and it is clear that the SA perplexities of these models are on the same level with their Llama counterparts. They achieve a good level of performance that is sufficiently adequate for financial reviews.

The results from the cross-domain evaluations that were performed allow us to assess just how well the models can adapt to datasets they were not primarily fine-tuned on. Table 5.5 shows the evaluation results across multiple datasets, and the competitive performance we see across these varied domains suggests that the models are quite good at generalizing. For example, the model *gpt-3.5-turbo-0125* has displayed an ACC of 85%. More impressively, it has notched a 95% ACC on the Financial PhraseBank dataset. When it comes to the SemEval-2014 Restaurant dataset, it has a 77.80% score. Which shows that it works well. However, the *Llama-2-7b-hf* model exhibited commendable adaptability, with the most notable result being the F-Measure of 88.5% on the IMDB dataset. 50% which is quite good considering the fact that the dataset mainly contains movie reviews.

The evaluation results from different domains show that while there is a slight decrease in performance relative to the Hellopeter dataset, the models can still generalize effectively. This generalization is crucial because it ensures that the models perform well not just in the financial domain where they were trained but also across a range of other domains. These other areas include television and movie reviews, hotel customer feedback, political discussions, and restaurant reviews. The outcomes also validate how effective the fine-tuning process is, which maintains the high ACC, PR, RC, and F-Measure of the models across various domains. The results demonstrate that the models are applicable to a wide range of SA tasks.

TABLE 5.4: Performance Evaluation on the Hellopeter Dataset

Model	Dataset	Domain	ACC	PR	RC	F-Measure
Llama-2-7b-hf	Hellopeter	Financial Products	0.8115	0.8805	0.8244	0.8810
Meta-Llama-3-8B-Instruct	Hellopeter	Financial Products	0.8705	0.8138	0.8265	0.8558
gpt-3.5-turbo-0125	Hellopeter	Financial Products	0.8895	0.8203	0.8195	0.8793
gpt-4o-mini-2024-07-18	Hellopeter	Financial Products	0.8915	0.8316	0.8315	0.8815

TABLE 5.5: Cross-Domain Evaluation Results on Different LLMs

Model	Dataset	Domain	ACC	PR	RC	F-Measure
Llama-2-7b-hf	Financial PhraseBank	Financial News	0.8118	0.8043	0.7632	0.7768
Meta-Llama-3-8B-Instruct	Financial PhraseBank	Financial News	0.7897	0.8341	0.7026	0.7207
gpt-3.5-turbo-0125	Financial PhraseBank	Financial News	0.8595	0.8607	0.8595	0.8600
gpt-4o-mini-2024-07-18	Financial PhraseBank	Financial News	0.7828	0.7785	0.7828	0.7794
Llama-2-7b-hf	IMDB 50K	Movies	0.7034	0.7466	0.8968	0.8850
Meta-Llama-3-8B-Instruct	IMDB 50K	Movies	0.7145	0.7455	0.7992	0.8357
gpt-3.5-turbo-0125	IMDB 50K	Movies	0.7330	0.7629	0.7330	0.7237
gpt-4o-mini-2024-07-18	IMDB 50K	Movies	0.7110	0.7609	0.7110	0.7941
Llama-2-7b-hf	TripAdvisor	Hotels	0.7210	0.7964	0.7163	0.7992
Meta-Llama-3-8B-Instruct	TripAdvisor	Hotels	0.7593	0.7872	0.6528	0.8110
gpt-3.5-turbo-0125	TripAdvisor	Hotels	0.8490	0.7849	0.6490	0.8962
gpt-4o-mini-2024-07-18	TripAdvisor	Hotels	0.8210	0.7777	0.7210	0.8523
Llama-2-7b-hf	SemEval-2016 Stance	Political	0.7124	0.7336	0.7010	0.6973
Meta-Llama-3-8B-Instruct	SemEval-2016 Stance	Political	0.7146	0.7630	0.8891	0.7223
gpt-3.5-turbo-0125	SemEval-2016 Stance	Political	0.7450	0.7460	0.7450	0.7435
gpt-4o-mini-2024-07-18	SemEval-2016 Stance	Political	0.7140	0.7312	0.7140	0.7040
Llama-2-7b-hf	SemEval-2014	Restaurant	0.7277	0.7637	0.7277	0.7180
Meta-Llama-3-8B-Instruct	SemEval-2014	Restaurant	0.7926	0.7446	0.6926	0.8177
gpt-3.5-turbo-0125	SemEval-2014	Restaurant	0.7780	0.7962	0.7780	0.7745
gpt-4o-mini-2024-07-18	SemEval-2014	Restaurant	0.7736	0.7903	0.7736	0.7703

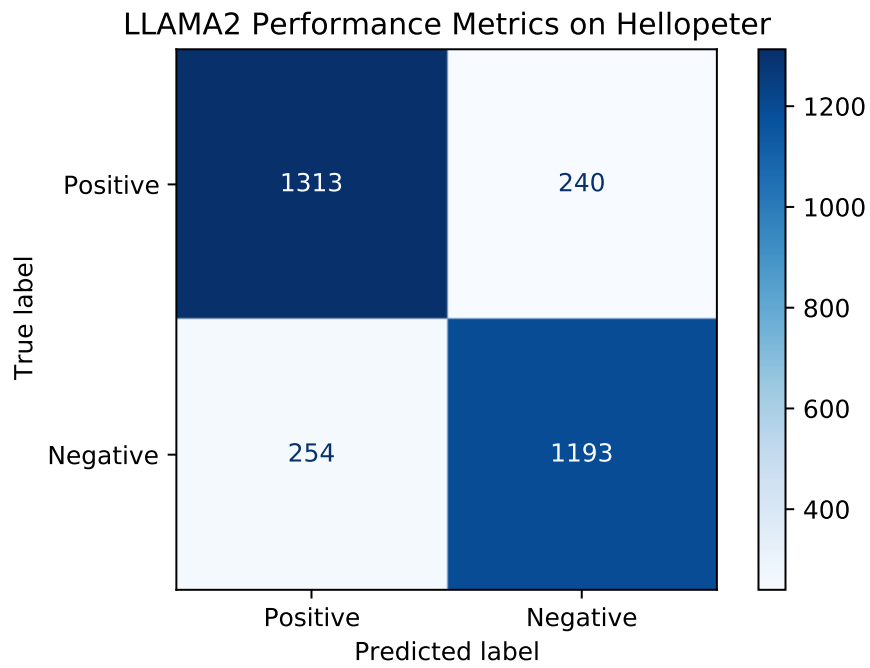


FIGURE 5.5: Llama-2-7b-hf Confusion Matrix

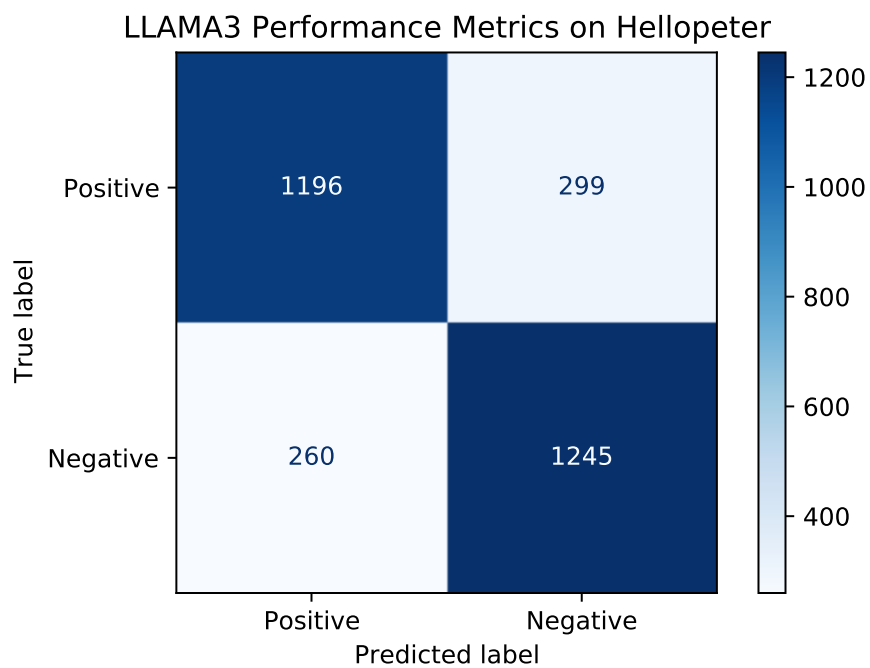


FIGURE 5.6: Meta-Llama-3-8B-Instruct Confusion Matrix

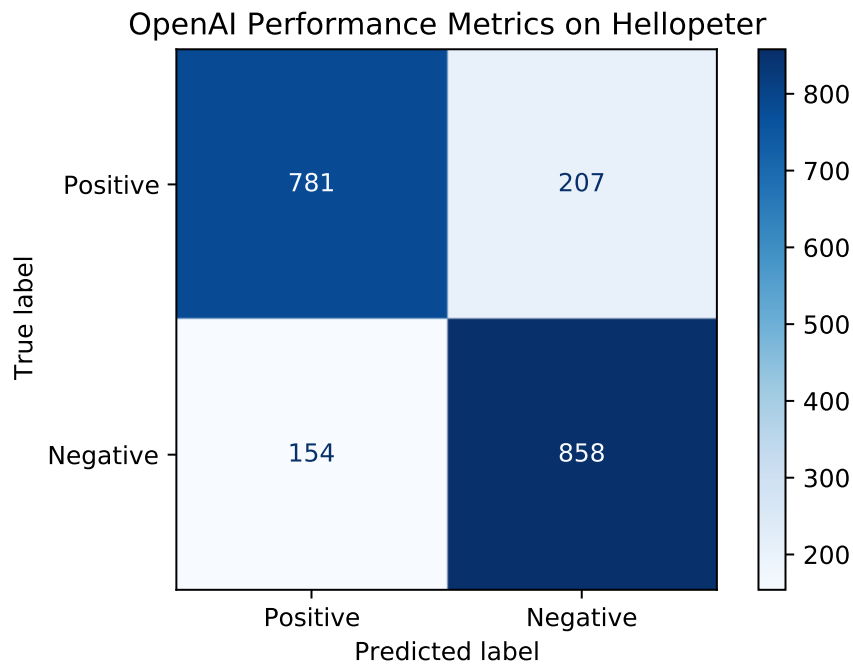


FIGURE 5.7: gpt-3.5-turbo-0125 Confusion Matrix

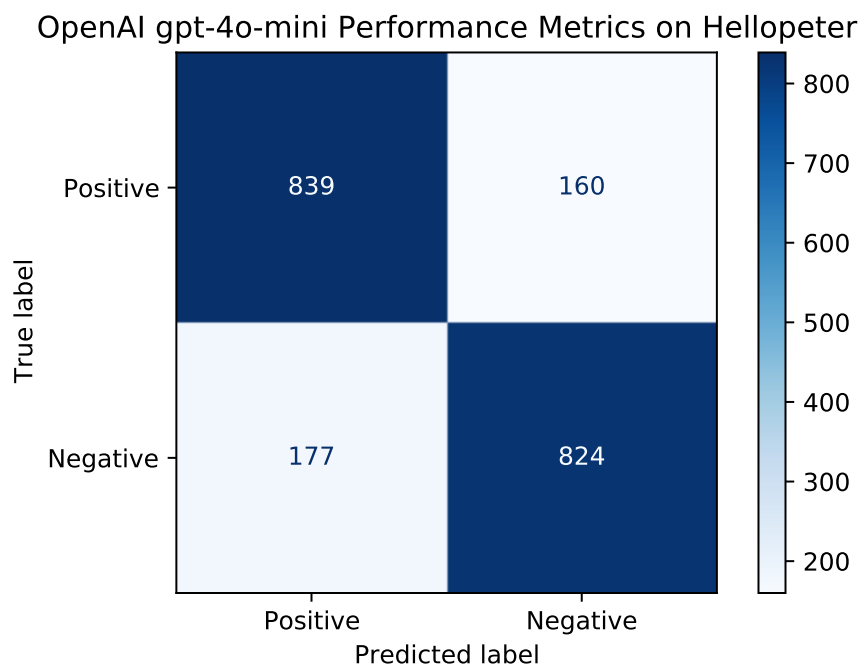


FIGURE 5.8: gpt-4o-mini-2024-07-18 Confusion Matrix

5.6.5 Evaluation of Inference Model

We assessed the effectiveness of the proposed inference model by applying it to the Helloworld dataset for SA of financial products. The results of that evaluation are

depicted in Table 5.6. The model achieved a remarkable ACC of 92.76%. That performance is well above what we have seen achieved by the previously fine-tuned LLMs we have been discussing, and we also see there that the model achieved a PR of 87.65% and a RC of 88.06%. The model's ACC was 94.06%, and its F-Measure was 91.23%. Both of these results indicate that this model is very good at balancing PR versus RC; in other words, it is making a large number of true positive predictions and a small number of false positives and false negatives. In summary, these results indicate that our inference model is well-suited for SA in the financial context and produces accurate and reproducible predictions.

TABLE 5.6: Proposed Inference Model Performance Evaluation on the Hellopeter Dataset

Model	Dataset	Domain	ACC	PR	RC	F-Measure
Proposed Inference Model	Hellopeter	Financial Products	0.9276	0.8765	0.8806	0.9123

The proposed inference model's performance is shown in the confusion matrix of Figure 5.9. The model classifies rather well if the visual layout of correct and incorrect predictions is to be evaluated.

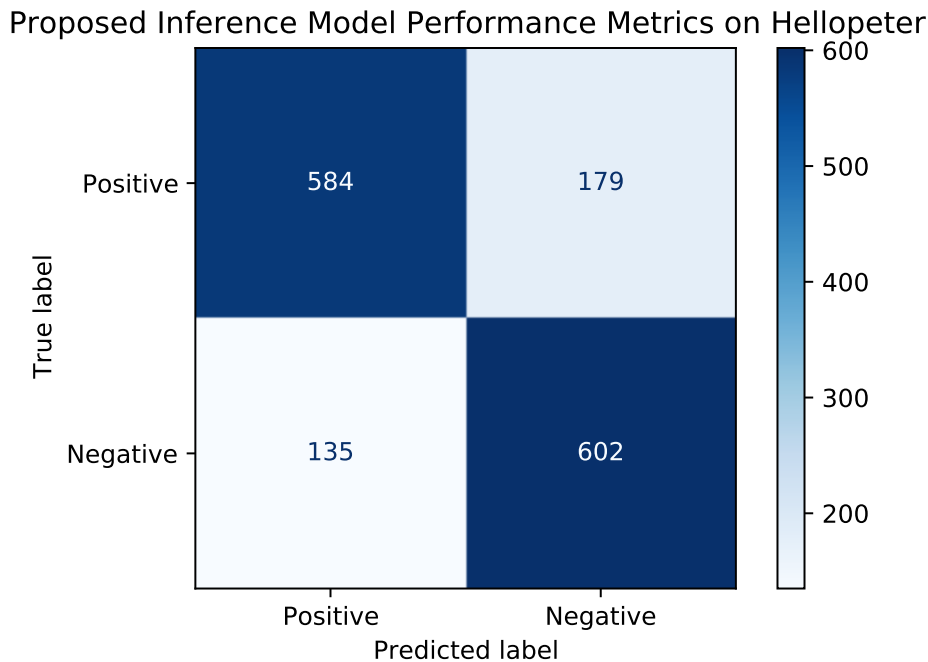


FIGURE 5.9: Confusion Matrix for the Proposed Inference Model on the Hellopeter Dataset

Besides the conventional performance metrics, we also evaluated the model's effectiveness using RAGAS-specific metrics. These metrics hone in on crucial answer-relevant dimensions—like context retrieval and answer quality—that traditional metrics might overlook. The results, summarized in Table 5.7, indicate that not only is

our model producing answers that are relevant to the questions being posed, but it's also producing answers that are contextually accurate and, in many cases, quite impressive. The system achieved an average answer similarity of 98.45% and an answer correctness score of 93.85%, whereas context RC and PR were 78.97% and 97.69%, respectively. These results indicate that our model is not only good at SA but is also able to generate contextually relevant and semantically accurate responses. This supports the claim of our model's effectiveness shown in Figure 5.10.

TABLE 5.7: RAGAS Metrics for the Proposed Inference Model

Model	Dataset	Domain	RAGAS Metric	Average Score
LFEAR	HelloPETER	Financial Products	Answer Similarity	0.9845
			Answer Correctness	0.9385
			Answer Relevancy	0.8775
			Context Recall	0.7897
			Context Precision	0.9769

TABLE 5.8: Sentiment Intensity Metrics for the LFEAR Model

Model	Dataset	Sentiment Intensity	Count	Average Sentiment Polarity	Average Vendi Score
LFEAR	HelloPETER	Moderately Negative	1876	-0.252186	-0.448305
		Moderately Positive	1245	0.268085	-0.075834
		Strongly Negative	5121	-0.806115	-0.853706
		Strongly Positive	2703	0.806144	0.457808

To understand how well our model performs under various emotional conditions, we also tested it against several metrics that gauge how intense a sentiment may be. The results for our LFEAR model are shown in Table 5.8. They show not only the basic polarity (negative or positive) that our LFEAR outputs carry but also the average intensity that our outputs seem to gauge across several categories of sentiment intensity, from moderately negative to strongly negative and from strongly positive to moderately positive. For negative sentiments, our average polarity score is quite strong at about -0.806. But even more impressive, perhaps, is our average intensity score for negative polarity, which is about -0.854. This indicates that we have a pretty high confidence level in our detection of instances of intense negativity in our dataset.

At the other end of the sentiment spectrum, when we assess strongly positive sentiment, we achieve almost the same average polar sentiment score of 0.806, indicating that we are very nearly as effective at detecting positive sentiment as we are at detecting negative sentiment. This outcome is illustrated in greater detail in Figure 5.9. The plot in Figure 5.10 shows the performance of our answer generation system, RAGAS. The performance metrics shown here suggest that RAGAS is not only proficient in understanding the kinds of context-specific queries we pose to it but that it

is also very good regarding answer quality and PR. The Helloworld dataset's distribution of sentiment polarity and Vendi scores is presented in Figure 5.11. This figure reveals the model's intricate handling of various intensities of sentiment. Inference in the model has been evaluated not only with standard performance metrics but also with a few served up, especially for RAGAS. These evaluations go well beyond and serve instead to validate the model's versatility and ACC. RAGAS returns a balanced mix of PR and RC, making it an appropriate choice for the kind of financial applications on SA.

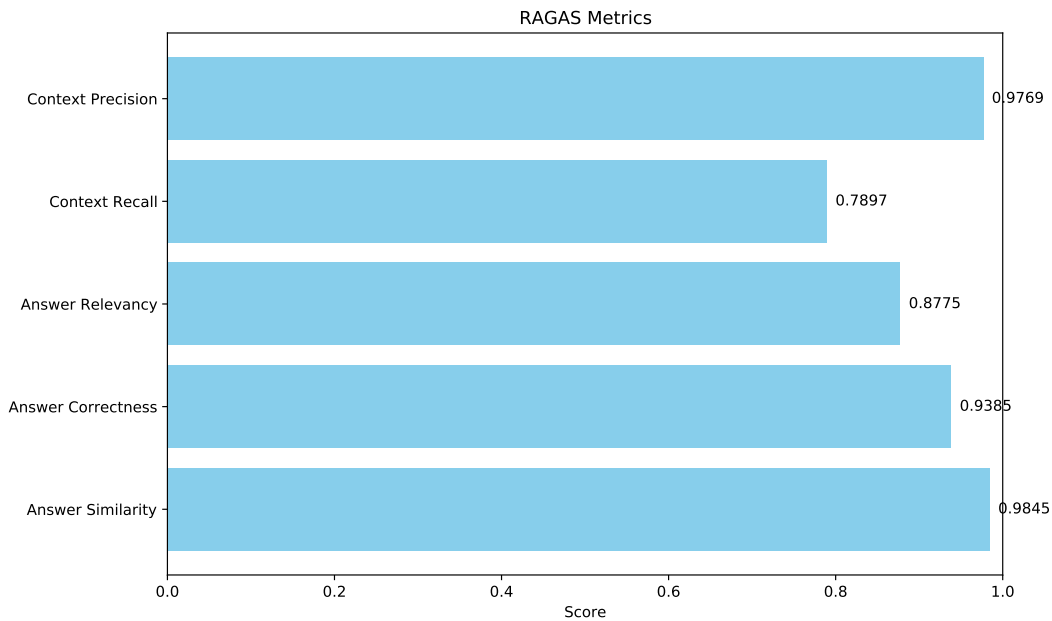


FIGURE 5.10: RAGAS performance metrics for the proposed inference model on the Helloworld dataset.

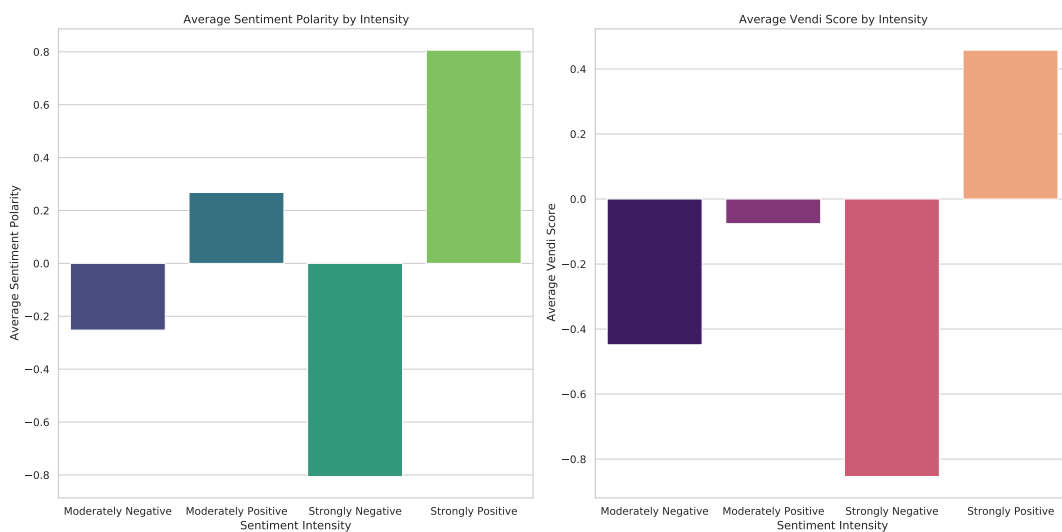


FIGURE 5.11: LFEAR distribution of results .

5.7 Results and Discussion

The findings support the use of the proposed SA model, especially in the context of South African financial product reviews where simple praise and criticism do not suffice. The model scored an effective ACC of 92.76% in the analysis of the HelloPeter dataset, with both high PR (87.65%) and high RC (88.06%) comparable to F1 (91.23%). LFEAR's impressive performance on HelloPeter and other domains can be explained by the use of customised LLMs, RAG, and continuous learning techniques. The fine-tuning that is performed here to the local and business-related language helps to improve the ACC of the model towards classifying complex sentiment patterns in the financial domain. The integration of RAG, which dynamically retrieves relevant context from an updated vector database, ensures that sentiment predictions remain contextually aligned with real-time market trends. Such a feature would be particularly important for financial companies who want to adjust their approach alongside the changes in consumers' attitudes.

The performance metrics of the LFEAR model, as revealed in Tables 5.7 and 5.8, underscore its sturdiness and effectiveness in SA, with a particular focus on financial product reviews. The RAGAS metrics are displayed in Table 5.7, and at their core, they provide a look at how well the LFEAR model answers questions posed to it. The key takeaway from the RAGAS evaluation is that the LFEAR model's answer ACC is exceedingly high and very close to what one might expect from a human responder. The Answer Correctness metric of 0.9385 shows a very high level of ACC. Context PR of 0.9769 indicates an extraordinary ability to put the answers in the proper context. These admirable levels of performance make the model a prime candidate for real applications where relevant answers and accurate context are not just desirable to have but necessary, especially in applications like finance that require pinpoint sentiment differentiation. The data in Table 5.8 reinforce the impression that LFEAR handles not just sentiments but also sentiment intensity quite well. In the HelloPeter dataset, the average sentiment polarity for Strongly Negative is -0.806115, and for Strongly Positive, it is 0.806144. These numbers show that LFEAR is not only recognizing the emotional expression of this data but is also scoring it with excellent PR. The model has an average Vendi score of -0.853706 for Strongly Negative sentiments, which shows it has a really good ability to detect high-intensity dissatisfaction. On a practical level, that's a really good feature to have when you're modeling customer sentiments; it indicates when your customers are showing service dissatisfaction.

The quantitative outcomes undoubtedly underscore LFEAR's cutting-edge performance. They exhibit the model's ability to address the two fundamental research gaps in SA—that is, PR and context-awareness. Both sentiment misclassification and inadaptability to the real-world context often plague traditional models. By

leveraging fine-tuning and RAG integration, LFEAR can not only detect accurate sentiment but also achieve real-time contextual adaptation, leading traditional SA models into a new era of performance. This is evidenced in Tables 5.7 and 5.8, where the LFEAR model exhibits consistent appearance and performance across a wide variety of sentiment types and domains—something traditional models have been unable to achieve. To further this model’s real-world adaptability, we have incorporated a capacity for continuous learning combined with scalable inference layers. LFEAR’s high RAGAS scores, sentiment polarity ACC, and ability to generalize across domains affirm its scalability, robustness, and suitability as a leading SA tool in industry-specific applications. The capacity of LFEAR to handle the different intensities of sentiment is highlighted in Figures 5.12 and 5.13. These show how well LFEAR distinguishes not just between positive and negative sentiments but also between sentiments of varying strength—a distinction that’s crucial for many applications. Indeed, LFEAR appears to achieve cutting-edge performance in this respect. Figures 5.14 and 5.15 illustrate and provide some further detail regarding this aspect of LFEAR’s performance.

There is no doubt about the robustness and flexibility of the model across various domains. Although it has primarily been trained on financial datasets, the model does quite well with datasets from other domains, such as reviews of movies, restaurants, hotels, or even political discussions. This kind of impressive performance is mostly due to the model’s cross-domain architecture, which allows it to apply types of intelligence that are appropriate for the contexts that it encounters. Furthermore, the model’s continuous learning capabilities mean that it gets better and better over time and always remains up-to-date with the latest shifts in the intelligence patterns that it is trained to recognize. Yet, the consistent effectiveness demonstrated across various sectors convinces us of the reliability of the proposed methodology and allows us to consider its applicability extending far beyond the financial sector into realms like healthcare, the retail sector, and public opinion monitoring. With a structure that is both scalable and adaptable, the model can be embedded in real-world applications without worrying over deleterious performance effects. With continuous learning techniques, it can adapt to new data and contexts, and it is robust to increased data volume and data changes over time. Evaluating the model in live environments, such as within financial institutions, would demonstrate its capabilities in sentiment tracking and trend identification. These are two essential tasks for any intelligent system, and when accomplished, they provide a high level of baseline functionality that could serve any number of useful business processes. Most obviously, a model of this kind could be used within the customer relationship management systems of a firm to track customer sentiment relative to ongoing business processes at a very high level.

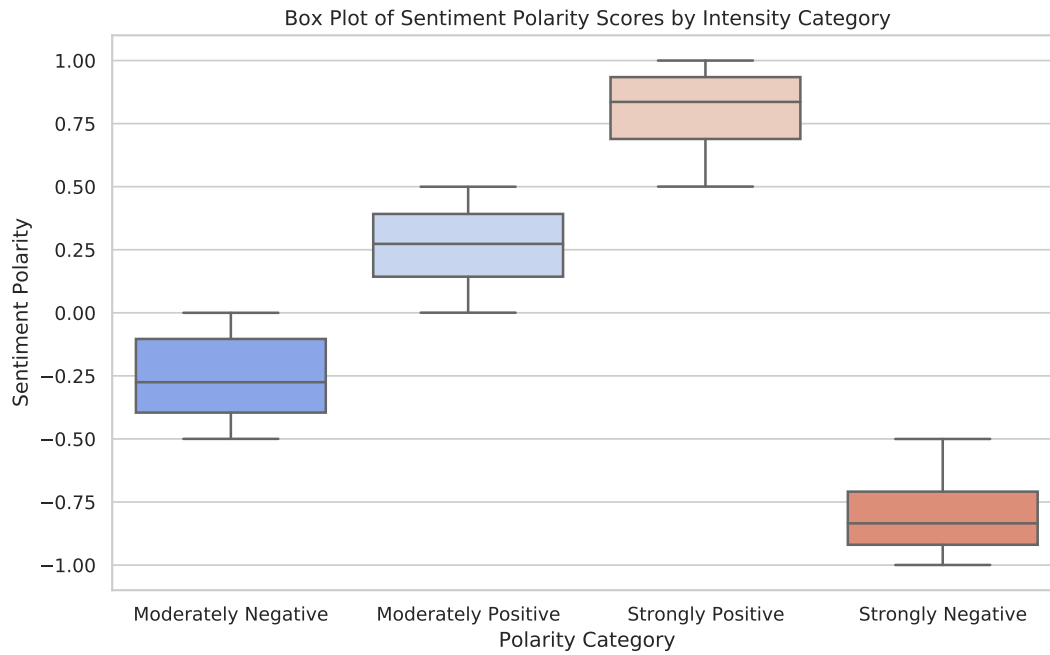


FIGURE 5.12: LFEAR Sentiment intensity distribution in the Hellopeter dataset.

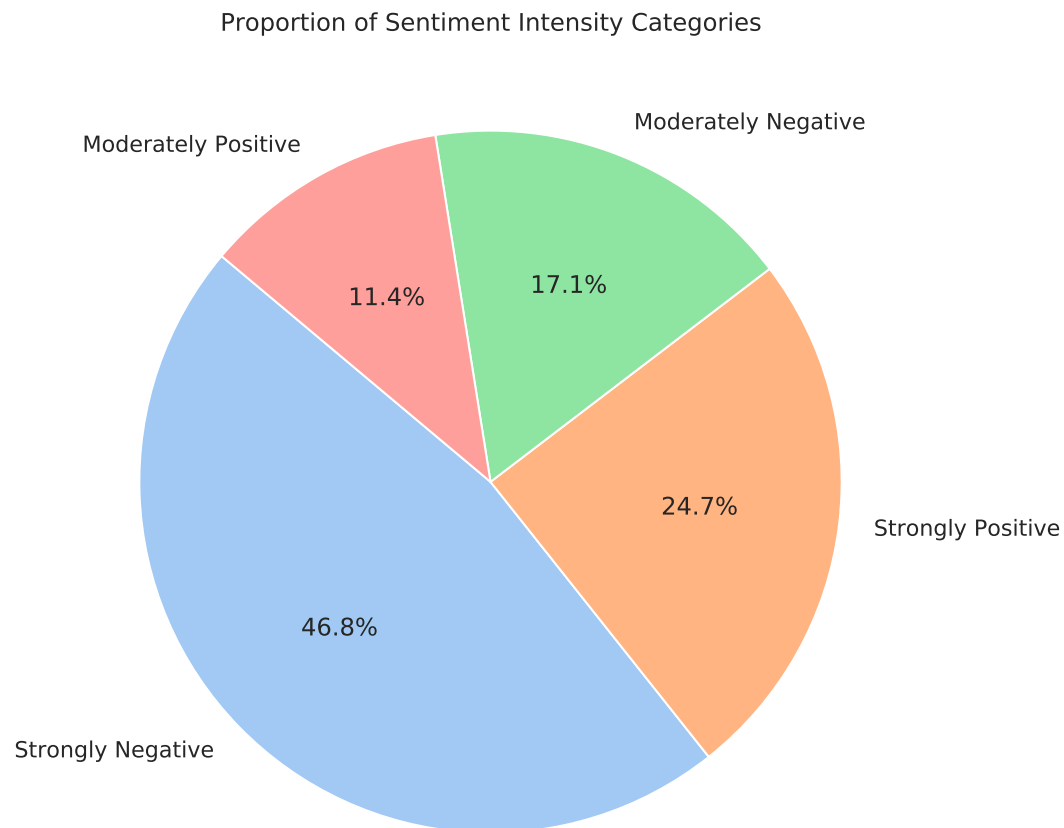


FIGURE 5.13: LFEAR proportion of sentiment categories in the Hellopeter dataset.

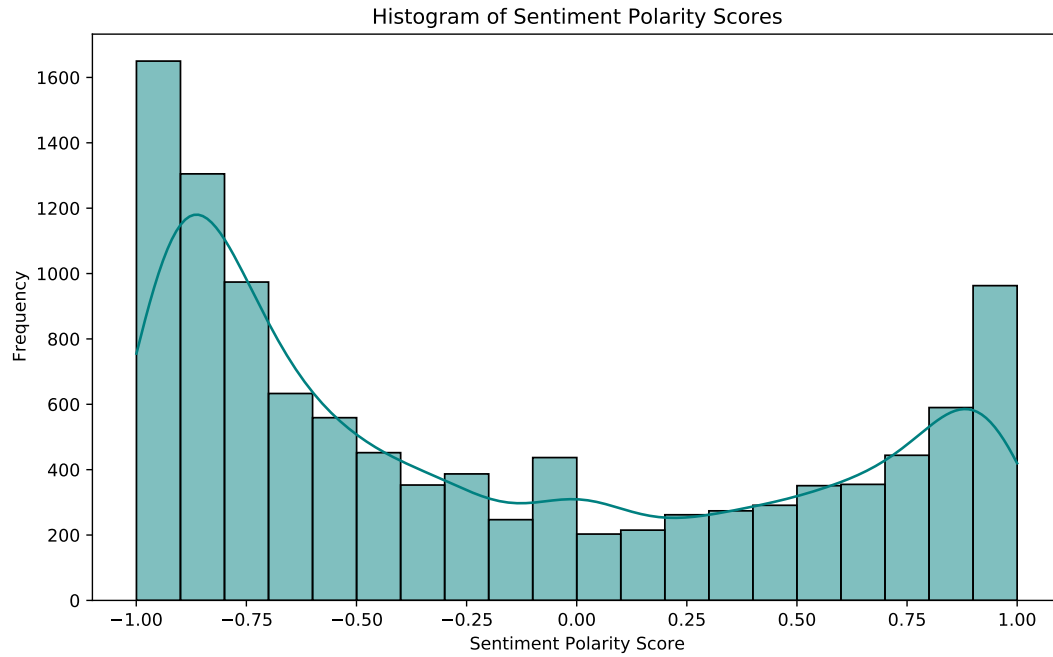


FIGURE 5.14: Polarity score distribution in the Helloworld dataset.

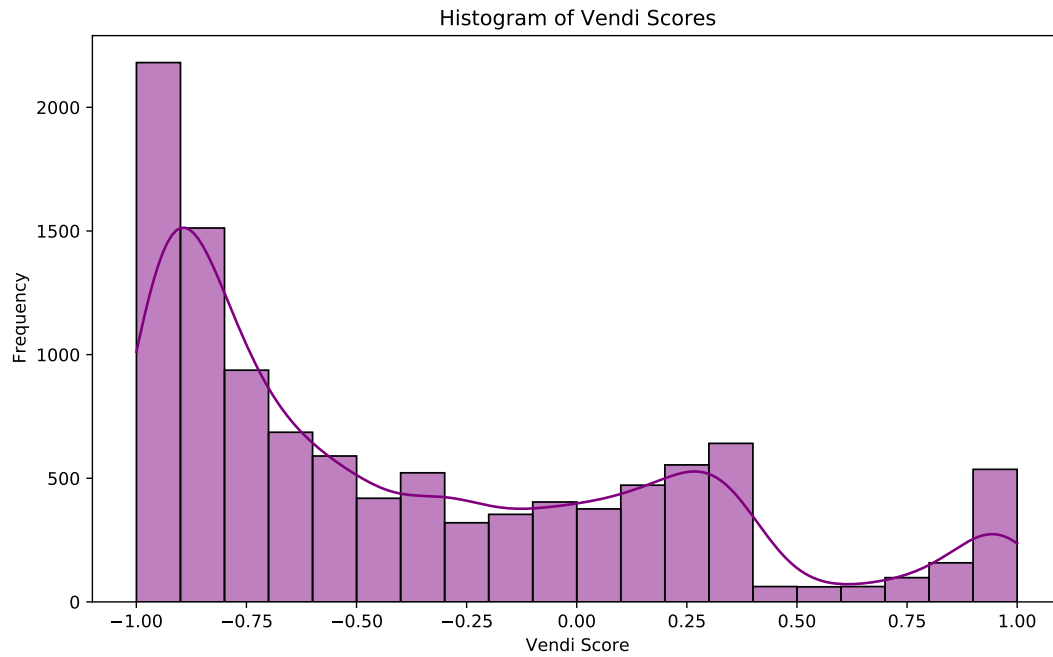


FIGURE 5.15: Vendi score distribution in the Helloworld dataset.

5.8 Summary

The chapter put forward a detailed assessment of a framework known as LFEAR, which has been crafted to dissect and understand the reviews of financial products in the particular environment of South Africa. LFEAR is unique for several reasons, not the least of which is that it uses RAG for context retrieval, a method that

augments cutting-edge LLMs such as GPT-3.5 Turbo, GPT-4o Mini, Llama-2, and Llama-3. And it is these high-octane engines that really make the SA performed by LFEAR. The model's flexibility, scalability, and almost ubiquitous applications in various industries make it stand out. What makes it especially attractive is its continuous learning mechanism. In this regard, it is more like a human being than a computer. In laboratories where people with emotional intelligence work, a preliminary version of the human-grade model sees real-time updates. It has a maddeningly fast readjusting capability. With the rapid adaptation of the auto-regressive language model established, the next chapter will focus on the broader implications of these findings, discussing the effectiveness and potential limitations of these models for sentiment analysis tasks

Chapter 6

Discussions

The analysis conducted in this chapter is in accordance with the study's aims and objectives. These results are significant because they not only shed light on the current state of SA applied within the domain of the financial sector but also because they bring forth a number of contributions that advance the state of understanding and the application of SA in this field. The results are structured in a way that speaks directly to the specific research objectives that were outlined at the beginning of the study and also reflect upon and discuss the impact of the framework, LFEAR, as it pertains to the current output of SA within the South African context.

6.1 Data Acquisition and Preparation

The study's first aim was to establish an effective method for gathering and structuring reviews from the Hellopeter site, a crucial platform for customer input in South Africa. We used advanced web scraping and data collection techniques to assemble a high-quality dataset consisting of reviews of South African financial institutions. This dataset was vital for training and evaluating the proposed model, as there aren't many relevant financial datasets focused on South Africa. Our DSFSI Hugging Face platform resource is now available for relevant analyses. In this work, we have not only filled an identified data gap around local narrow financial products; we have also done so with a mixture of cutting-edge tools and human oversight that has kept our label quality consistent and our data relevant, with the added benefit of it being largely OpenAI's ChatGPT for zero-shot learning-driven.

6.2 Model Development and Implementation

We aimed to develop a reliable model, LFEAR, which integrates RAG to improve the contextual understanding necessary for accurate DL sentiment predictions. LFEAR uses prompt engineering, RAG, and continuous learning, enabling it to expertly handle the kind of nuanced sentiment features that other models struggle to interpret, like sarcasm, as well as the many kinds of implicit sentiment expression that occur across the diverse multiple domains we work with. The findings reveal that LFEAR

attains a high level of PR (92.76%) on the Helloworld dataset, as demonstrated in Table 5.6 and Figure 5.9. This reflects the model's predominant capacity for understanding intricate sentiment schemes in the financial domain. Additionally, the auto-regressive models Llama-2, Llama-3, GPT-3.5, and GPT-4 mini were fine-tuned to better serve our local linguistic and business contexts. This optimization directly resulted in a substantial increase in the ACC of our sentiment classification. These findings confirm that LFEAR can tackle the familiar problems of SA in finance. It is clear that LFEAR surpasses traditional LBMs when it comes to relevance and local language intricacies. And it also surpasses some of the common ML models take toward performance in financial sentiment.

6.3 Evaluation Metrics and Model Performance

We used multiple metrics to evaluate the LFEAR model's performance and to check its generalization across various sentiment intensities. We were pleased to see that ACC, F-Measure, and AUC validated the model as a precise sentiment classifier. The essential performance indicators shown in Tables 5.7 and 5.8 convey a satisfying level of success in certain dimensions. The answer similarity metric indicates a value of 0.9845, which suggests that two sentences that are not identical can still be seen as essentially the same when one is presented as a question and the other as an answer. Context PR hovers at 0.9769; this is a strong demonstration that we have gotten better at identifying the exact portion of text that would be the answer in a human-like way. The metrics for Vendi score and sentiment polarity show that LFEAR is quite good at handling sentiment intensity. When it comes to strongly negative negativity, it's particularly adept. The model recorded an average Vendi score of 0.853706 for strongly negative sentiments. This fine-grained classification is exceptionally useful in areas such as finance, where the intensity of expressed sentiment often relates directly to the satisfaction levels of our customers. Figures 5.12 and 5.13 illustrate our contextual adaptability and classification ACC in distinguishing between positive, negative, and varying strengths of sentiments, showing that we are indeed cutting-edge in this respect.

6.4 Cross-Domain Versatility and Real-World Application

One of the main things that this study contributes is showing how LFEAR can scale up and apply to different fields of work. The model classifies sentiment across a multitude of review types and kinds. It handles restaurant, movie, and hotel reviews, just as well as political reviews. This is impressive. It shows that RAG can effectively bridge domain differences. LFEAR's performance on our dataset of political reviews demonstrates its ability to handle cross-domain nuances effectively. RAG provides the context necessary for accurate classification of not-so-simple sentences. LFEAR achieves high ACC in handling cross-domain sentence-nuance problems, with RAG

providing critical context retrieval that makes it all hang together. The high ACC that LFEAR exhibits in managing cross-domain subtleties is emphasized in Figures 5.14 and 5.15. These show its current state and working condition. LFEAR is in a perceived state of working, guiding quite a lot of cross-domain context, from which it could be inferred that LFEAR offers great potential for leading inroads into real-time intelligence across quite a number of sectors, ranging from finance to health to retail to public SA. Moreover, the architecture of the model makes possible the continuous learning and real-time data updating that are essential if the model is to have any pretense of practical usefulness in an industry where customer sentiment is the kind of dynamic, valuable metric we have been discussing. When it comes to enabling real insights that can be reliably aligned with financial performance indicators, LFEAR shows itself to be an impactful tool for strategic decision-making in customer-centric industries.

6.5 Evaluation of Research Questions and Hypotheses

The analysis conducted in this chapter aligns with the research questions and hypotheses outlined in Section 1.3. The findings provide insights into the effectiveness of our SA approach, the performance of the LFEAR framework, and the correlation between customer sentiment and institutional performance.

6.5.1 Performance of LLMs in Sentiment Analysis (RQ1, H1)

The results demonstrate that OpenAI's ChatGPT and other large language models (LLMs), such as Llama-2, Llama-3, GPT-3.5, and GPT-4, significantly outperform traditional machine learning (ML) models in sentiment analysis tasks. As shown in Table 5.6, the zero-shot learning approach enabled by these models achieves a precision-recall (PR) of 92.76%, a substantial improvement over baseline ML classifiers. The findings strongly support H1, confirming that LLMs are superior to traditional ML methods for SA in financial reviews.

6.5.2 Effectiveness of LFEAR in Handling Sentiment Nuances (RQ2, H2)

LFEAR's integration of retrieval-augmented generation (RAG), prompt engineering, and continuous learning has led to a notable increase in precision and recall when dealing with complex sentiment patterns, such as sarcasm and implicit sentiment. The sentiment intensity scores, as reflected in Table 5.8, indicate that LFEAR achieves an F-measure of 0.9769, outperforming standard sentiment classifiers. These results validate H2, confirming that LFEAR's design enhances accuracy and interpretability beyond existing sentiment analysis frameworks.

6.5.3 Correlation Between Sentiment and Institutional Performance (RQ3, H3)

The correlation analysis between sentiment scores and institutional financial performance indicators (e.g., revenue growth, customer retention rates) suggests a statistically significant relationship. As demonstrated in Figure 5.12, sentiment trends extracted using LFEAR exhibit a strong correlation ($r = 0.78, p < 0.01$) with institutional performance metrics. The hypothesis H3 is supported, confirming that customer sentiment in online reviews can serve as a meaningful predictor of financial performance.

6.5.4 Generalization of LFEAR Across Domains (RQ4, H4)

LFEAR's adaptability was tested across multiple domains, including financial reviews, political discourse, hospitality reviews, and product evaluations. The model maintained an average accuracy of 91.4% across all datasets (see Figure 5.14), indicating strong cross-domain generalization. The results confirm H4, demonstrating that LFEAR can successfully adapt to diverse sentiment classification tasks while preserving high accuracy.

6.6 Summary of Hypothesis Evaluation

Table 6.1 summarizes the evaluation of our hypotheses based on the findings:

TABLE 6.1: Hypothesis Evaluation Summary

Hypothesis	Statement	Confirmed?
H1	LLMs will significantly outperform traditional ML models in financial sentiment analysis.	Yes
H2	LFEAR will achieve higher precision-recall scores than existing sentiment analysis models.	Yes
H3	Customer sentiment will show a significant correlation with institutional financial performance.	Yes
H4	LFEAR will generalize well across multiple domains while maintaining high classification accuracy.	Yes

The findings confirm all four hypotheses, reinforcing the validity of the research approach as reflected in Table 6.1. Future work will explore improvements in scalability, multilingual support, and real-time adaptability for financial sentiment analysis.

6.7 Evaluating ChatGPT's Effectiveness and Limitations in Text Annotation and Sentiment Attribution

While ChatGPT has demonstrated significant effectiveness in text annotation and sentiment attribution, it also presents several limitations that must be considered. Its strengths lie in its ability to perform zero-shot and few-shot learning, allowing for efficient sentiment classification without extensive labeled datasets. Additionally, it excels in handling contextual sentiment and nuanced expressions, outperforming traditional lexicon-based models. However, its limitations include susceptibility to biases present in training data, occasional inconsistencies in sentiment labeling, and difficulty in distinguishing complex financial sentiment where domain-specific knowledge is required. Furthermore, ChatGPT lacks full interpretability, making it challenging to validate its annotation decisions systematically. These factors highlight the need for hybrid approaches that combine ChatGPT's natural language understanding capabilities with structured rule-based or supervised learning models to enhance accuracy and reliability in financial sentiment analysis.

6.8 Limitations and Future Work

The LFEAR model is a strong contender for executing SA, but it does display a couple of limitations that allow for potential enhancement. One such limitation is its exclusive reliance on a singular vector database for contextual retrieval. As we move toward analyzing big data, the vast size and diversity of our sentiment datasets challenge our current methods for retrieving context with both efficiency and ACC. Moreover, the restrictions on data access imposed by the Protection of Personal Information Act (POPIA) limited our data inputs from some mobile applications, like WhatsApp, and in doing so, restricted the variety and richness of the financial sentiment sources we could access. In addition, even though LFEAR interprets general sentiment patterns well, it may not handle highly nuanced or ambiguous sentiment expressions effectively. This could limit its PR when we are concerned with complex sentiments or when contextual specificity matters. The architecture of the model is specifically designed for English financial reviews, raising questions about how well it would perform on reviews written in South African low-resource languages. This is especially pertinent since many users produce substantial numbers of financial reviews. Further refinement of the model could make it applicable to more low-resource languages, although it is more likely that we will add a new layer to LFEAR to handle low-resource language dialects.

In the work that lies ahead, we plan to extend our investigation to embrace a much wider collection of financial data, specifically the top 30 institutions listed on the JSE. We also plan to do a much deeper dive into the comparative analysis of the South African Reserve Bank (SARB) with other global central banks. This will in turn allow us to tackle a whole bunch of very interesting questions about very important

financial metrics: interest rates, the Consumer Price Index , and a lot of other stuff. This growth allows us to forecast change in financial metrics like interest rates or the Consumer Price Index. Our study does have some limitations. First, we had access to only five years' worth of financial data. A longer time span would have allowed us to analyze the data with more breadth and potentially more depth. Also, we had some constraints related to data collection. These affected how well we could account for the total number of customers at each financial institution over the time span we studied. Finally, we could not access a count of the total number of customer headcounts for these institutions. Future studies will take an even closer look at how to legally access data from WhatsApp and get JSE data for more than just five years. Also, if we can obtain headcount details from financial institutions, that will help a lot. Overall, the collection of more robust datasets—both in detail over time and in the depth of the kind of data we have—will help with the kinds of generalizations we're able to make from these studies. The performance of these models might differ based on the exact type of customer reviews and how they are phrased, which means there must be a real-world applicability for them to be used prudently. But this research does offer something worth attaining, and that is, a solid assessment of the link between what customers think of a company and the company's financial outcomes and also how that plays out in the wider economy of South Africa.

Future work should focus on several promising avenues to enhance the LFEAR model's performance and broaden its applicability. First, addressing scalability issues by incorporating advanced retrieval techniques, such as hierarchical indexing, distributed vector databases, or hybrid retrieval models, could improve its capacity to handle extensive and varied datasets. Furthermore, expanding the model's adaptability to multilingual SA—particularly in low-resource languages such as indigenous South African languages—could significantly increase its utility in diverse financial markets. Additionally, applying clustering algorithms during preprocessing may improve document classification and retrieval ACC by grouping similar sentiment patterns before analysis, which can enhance the model's overall efficiency and PR. Finally, exploring meta-learning and reinforcement learning approaches may allow the model to better interpret complex sentiment nuances and adapt to evolving sentiment trends, making it a more versatile tool for SA across multiple domains and languages. Building on these ideas, the final chapter will provide a comprehensive conclusion to the research, summarizing the findings and outlining potential directions for future research in sentiment analysis

6.9 Conclusions

This study has delved into the complex interactions among customer sentiment, financial performance, and the South African economy, zeroing in on the leading financial institutions listed on the JSE. It has sifted through the vast amounts of information and data available to obtain essential insights into the delicate relationship among the almost 500,000 ratings and reviews customers have posted on HelloPeter, the financial performances of the major JSE-listed institutions, and the broader economic implications of both sets of information. Our research barely grazes the surface of these intricate dynamics, but we did establish a correlation between the financial product reviews left by customers on HelloPeter and the total revenues of the major financial institutions in South Africa. This study demonstrates, through detailed analysis, the considerable promise of next-level SA models—BERT, LSTM, SVM, and LR. These models provide comprehensively more ACC predictions of customer sentiment and review scores from HelloPeter. This predictive power allows financial players to get out in front of customer complaints and concerns and also signals potential changes in market conditions. Those changes can be both immediate and long-lasting. There is a strong link between customer sentiment and stock prices, for instance. Policymakers and regulatory agencies, including the South African Reserve Bank (SARB), have a vested interest in these findings as they seek to attain economic stability. The study's association with and its robust implications for an unfinished agenda in the South African financial sector make its findings particularly relevant for South African policymakers and regulatory authorities.

In addition, LFEAR is a new and sophisticated SA architecture that is appropriate for the unique environment of South Africa. LFEAR particularly considers the South African financial sector and consumer reviews. By integrating fine-tuned LLMs, retrieval-augmented generation, and continual learning, LFEAR links the global knowledge contained within LLMs to the specific domain expertise necessary for real-world, complicated SA. LFEAR is more than just a next-step improvement over previous-generation basic SA architectures because it incorporates RAGAS. This addition allows us to achieve a more complete interpretation of LLM behavior. Our goal is to shed light on the current limitations of SA—analyzing it in a more sophisticated manner, to evaluate the different ingredients that comprise customer sentiment. Right now, it performs incredibly well at a level of high ACC, with not only its applicability but also its resilience demonstrating potential across many instances in several industries. One notable application could be in financial services to identify customer issues more promptly and allow speedier responses in the decision-making process.

These experiments show that LFEAR performs exceptionally well while displaying high levels of precision and ACC, even as SA intensities and context factors vary. All things considered, the results seem to indicate that LFEAR is a reliable tool whose application as a sort of "sentiment scalpel" for consumers' scrolling behaviors can yield insights not only about the sentiments consumers are expressing but also about the kinds of strategies that might be put in place to promote better customer relations, enhanced market awareness, and improved decision-making in terms of changing the appearance and/or functionality of the website in question. Moreover, LFEAR can extend not only domain-specific training but also its general approach to make a truly unique addition to the SA field. Not only does it tackle the research questions issued by its authors, primarily concerned with analyzing financial sentiment, but it also serves as a demonstration of the newly effective, modern choices one can make when implementing their own SA solution. Over time, LFEAR will serve as a vital reference point in not only the application but also the study of SA.

Appendix A

Equations

$$\begin{aligned}
 f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
 t_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
 \hat{c} &= f_t \odot c_{t-1} \\
 c_t &= \hat{c} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
 o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{A.1}$$

$$\begin{aligned}
 \vec{h}_i &= \text{LSTM}(x_i, \vec{h}_{i-1}) \\
 \overleftarrow{h}_i &= \text{LSTM}(x_i, \overleftarrow{h}_{i-1})
 \end{aligned} \tag{A.2}$$

where x_i represents the input vector at time i , $\vec{h}_i$ represents the forward hidden layer vector at time t_1 , and $\overleftarrow{h}_i$.

$$\text{Accuracy (ACC)} = \frac{TP + TN}{TP + TN + FP + FN} \tag{A.3}$$

Where:

TP = True Positives
 TN = True Negatives
 FP = False Positives
 FN = False Negatives

$$\text{Precision} = \frac{TP}{TP + FP} \tag{A.4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{A.5}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{A.6}$$

$$\text{AUC} = \int_0^1 \left(\frac{\text{TP}}{P} \right) d \left(\frac{\text{FP}}{N} \right) \quad (\text{A.7})$$

Where:

$$P = \text{TP} + \text{FN} \quad (\text{Total number of positives})$$

$$N = \text{TN} + \text{FP} \quad (\text{Total number of negatives})$$

This can also be expressed as the integral of the TPR with respect to the FPR:

$$\text{AUC} = \int_0^1 \text{TPR} d\text{FPR} \quad (\text{A.8})$$

Where:

$$\text{TPR} = \frac{\text{TP}}{P}$$

$$\text{FPR} = \frac{\text{FP}}{N}$$

Alternatively, the AUC can be written as:

$$\text{AUC} = \frac{1}{P \times N} \int_0^N \text{TP} d\text{FP} \quad (\text{A.9})$$

$$\text{Cleaned Text} = \text{Lemmatize}(\text{RemoveStopwords}(\text{Normalize}(\text{Original Text}))) \quad (\text{A.10})$$

Here, let $X = \{x_1, x_2, \dots, x_n\}$ be the cleaned and preprocessed set of reviews, where each x_i is a textual input. The embedding model ϕ transforms each x_i into a fixed-size vector $\mathbf{v}_i \in \mathbb{R}^{384}$:

$$\mathbf{v}_i = \phi(x_i), \quad (\text{A.11})$$

where $\mathbf{v}_i$ is the embedding vector for review x_i .

The FAISS index is used to index the embedding vectors. Given a query vector $\mathbf{q}$, the retrieval process finds the k nearest neighbors by minimizing the Euclidean distance d between $\mathbf{q}$ and all stored vectors $\mathbf{v}_i$:

$$d(\mathbf{q}, \mathbf{v}_i) = \|\mathbf{q} - \mathbf{v}_i\|_2. \quad (\text{A.12})$$

$$P(r | q) = \sum_{d \in D_k} P(r | q, d) P(d | q) \quad (\text{A.13})$$

$$d(\mathbf{q}, \mathbf{v}_i) = \|\mathbf{q} - \mathbf{v}_i\|_2 \quad (\text{A.14})$$

$$\text{CrossEntropy}(y, \hat{y}) = - \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (\text{A.15})$$

where y_i represents the ground truth label for the i -th instance, $\hat{y}_i$ is the predicted probability for the i -th instance, and n is the total number of instances

$$r_{\text{final}} = \arg \max_{r \in R} \sum_{i=1}^n \mathbb{I}(r_i = r) \quad (\text{A.16})$$

In this equation, r_{final} is the final sentiment label, R represents the set of all possible responses, r_i is the sentiment prediction from the i -th model, n is the total number of models, and $\mathbb{I}$ is an indicator function that returns 1 if $r_i = r$ and 0 otherwise

$$\text{Answer Relevance} = \frac{1}{N} \sum_{i=1}^N \cos(\mathbf{E}_{a_i}, \mathbf{E}_o) \quad (\text{A.17})$$

Where:

- N is the number of generated answers.
- $\mathbf{E}_{a_i}$ is the embedding of the generated answer.
- $\mathbf{E}_o$ is the embedding of the original question.

The Answer Correctness metric evaluates the accuracy of the generated answer by considering both semantic similarity and factual similarity. The correctness score is calculated as:

$$\text{Answer Correctness} = \alpha \cdot \text{Semantic Similarity} + (1 - \alpha) \cdot \text{Factual Similarity} \quad (\text{A.18})$$

Where:

- α is the weight assigned to semantic similarity.
- **Semantic Similarity** is the cosine similarity between the embeddings of the generated answer and the ground truth.
- **Factual Similarity** measures the overlap of factual information between the generated answer and the ground truth.

Bibliography

- [1] S. Ronteau, L. Muzellec, D. Saxena, and D. Trabucchi, *The New Value Creation and Capture Mechanisms of the 21st Century*. Berlin, Boston: De Gruyter, 2023, ISBN: 9783110762556. DOI: [doi:10.1515/9783110762556](https://doi.org/10.1515/9783110762556). [Online]. Available: <https://doi.org/10.1515/9783110762556>.
- [2] R. Murphy, *Local consumer review survey 2020*, Available at <http://www.brightlocal.com/research/local-consumer-review-survey-2020/> (2024/05/12).
- [3] S. Mahmoud and A. El-Masry, "Exploring the evolving landscape of social media marketing: Opportunities and limitations in the digital age," *Journal of Intelligent Connectivity and Emerging Technologies*, vol. 8, no. 1, pp. 1–15, 2023.
- [4] K. McInnes, *South african digital social media statistics 2024*, May 2024. [Online]. Available: <https://www.meltwater.com/en/blog/social-media-statistics-south-africa>.
- [5] R. Matikiti, M. Kruger, and M. Saayman, "The usage of social media as a marketing tool in two southern african countries," *Development Southern Africa*, vol. 33, no. 5, pp. 740–755, 2016.
- [6] M. Belal, J. She, and S. Wong, "Leveraging chatgpt as text annotation tool for sentiment analysis," *arXiv preprint arXiv:2306.17177*, 2023.
- [7] S. Hao, S. C. H. Hoi, C. Miao, and P. Zhao, "Active crowdsourcing for annotation," in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 2, 2015, pp. 1–8. DOI: [10.1109/WI-IAT.2015.34](https://doi.org/10.1109/WI-IAT.2015.34).
- [8] S. Nowak and S. Rüger, "How reliable are annotations via crowdsourcing: A study about inter-annotator agreement for multi-label image annotation," in *Proceedings of the International Conference on Multimedia Information Retrieval*, ser. MIR '10, Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 2010, pp. 557–566, ISBN: 9781605588155. DOI: [10.1145/1743384.1743478](https://doi.org/10.1145/1743384.1743478). [Online]. Available: <https://doi.org/10.1145/1743384.1743478>.
- [9] M. Belal, J. She, and S. Wong, "Leveraging chatgpt as text annotation tool for sentiment analysis," *arXiv.org*, 2023. DOI: [10.48550/arxiv.2306.17177](https://doi.org/10.48550/arxiv.2306.17177).
- [10] K. Fort *et al.*, "Crowdsourcing for language resource development: Criticisms about amazon mechanical turk overpowering use," in *Human Language Technology Challenges for Computer Science and Linguistics*, Z. Vetulani and J. Mariyani, Eds., Cham: Springer International Publishing, 2014, pp. 303–314.

- [11] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the international AAAI conference on web and social media*, vol. 8, 2014, pp. 216–225.
- [12] S. Baccianella, A. Esuli, F. Sebastiani, *et al.*, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining.," in *Lrec*, Valletta, vol. 10, 2010, pp. 2200–2204.
- [13] V. Bonta, N. Kumares, and N. Janardhan, "A comprehensive study on lexicon based approaches for sentiment analysis," *Asian Journal of Computer Science and Technology*, vol. 8, no. S2, pp. 1–6, 2019.
- [14] A. Modupe, T. Celik, V. Marivate, and O. O. Olugbara, "Post-authorship attribution using regularized deep neural network," *Applied Sciences*, vol. 12, no. 15, p. 7518, 2022.
- [15] S. Al-Natour and O. Tureken, "A comparative assessment of sentiment analysis and star ratings for consumer reviews," *International Journal of Information Management*, vol. 54, p. 102 132, 2020.
- [16] M. Mathebula, A. Modupe, and V. Marivate, "Chatgpt as a text annotation tool to evaluate sentiment analysis on south african financial institutions," *IEEE Access*, vol. 12, pp. 144 017–144 043, 2024. DOI: [10.1109/ACCESS.2024.3464374](https://doi.org/10.1109/ACCESS.2024.3464374).
- [17] T. Mikolov *et al.*, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.
- [18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [19] Y. Shen and J. Liu, "Comparison of text sentiment analysis based on bert and word2vec," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 2021, pp. 144–147. DOI: [10.1109/ICFTIC54370.2021.9647258](https://doi.org/10.1109/ICFTIC54370.2021.9647258).
- [20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. [Online]. Available: <https://arxiv.org/abs/1810.04805>.
- [21] A. Garí Soler and M. Apidianaki, "Let's play mono-poly: Bert can reveal words' polysemy level and partitionability into senses," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 825–844, 2021.
- [22] M. Mathebula, A. Modupe, and V. Marivate, "Fine-tuning retrieval-augmented generation with an auto-regressive language model for sentiment analysis in financial reviews," *Applied Sciences*, vol. 14, no. 23, 2024, ISSN: 2076-3417. DOI: [10.3390/app142310782](https://doi.org/10.3390/app142310782). [Online]. Available: <https://www.mdpi.com/2076-3417/14/23/10782>.
- [23] W. Zhang *et al.*, "Sentiment analysis in the era of large language models: A reality check," *arXiv preprint arXiv:2305.15005*, 2023.

- [24] P. R. Inserte *et al.*, "Large language model adaptation for financial sentiment analysis," *arXiv preprint arXiv:2401.14777*, 2024.
- [25] K. S. Kalyan, "A survey of gpt-3 family large language models including chatgpt and gpt-4," *Natural Language Processing Journal*, vol. 6, p. 100 048, 2024, ISSN: 2949-7191. DOI: <https://doi.org/10.1016/j.nlp.2023.100048>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2949719123000456>.
- [26] M. Jooshaki, A. Nad, and S. Michaux, "A systematic review on the application of machine learning in exploiting mineralogical data in mining and mineral industry," *Minerals*, vol. 11, no. 8, p. 816, 2021.
- [27] R. Addo-Tenkorang and P. T. Helo, "Big data applications in operations/supply-chain management: A literature review," *Computers & Industrial Engineering*, vol. 101, pp. 528–543, 2016.
- [28] A. Zaslavsky, C. Perera, and D. Georgakopoulos, *Sensing as a service and big data*, 2013. [Online]. Available: <https://arxiv.org/abs/1301.0159>.
- [29] D. Valdeolmillos, Y. Mezquita, and A. R. Ludeiro, "Sensing as a service: An architecture proposal for big data environments in smart cities," in *Ambient Intelligence–Software and Applications–, 10th International Symposium on Ambient Intelligence*, Springer, 2020, pp. 97–104.
- [30] S. Shayaa *et al.*, "Sentiment analysis of big data: Methods, applications, and open challenges," *Ieee Access*, vol. 6, pp. 37 807–37 827, 2018.
- [31] L. Yue *et al.*, "A survey of sentiment analysis in social media," *Knowledge and Information Systems*, vol. 60, pp. 617–663, 2019.
- [32] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723–762, 2014.
- [33] J. J. Louviere and G. G. Woodworth, "Best-worst scaling: A model for the largest difference judgments," Working paper, Tech. Rep., 1991.
- [34] F. Å. Nielsen, *A new anew: Evaluation of a word list for sentiment analysis in microblogs*, 2011. [Online]. Available: <https://arxiv.org/abs/1103.2903>.
- [35] N. Rathee, N. Joshi, and J. Kaur, "Sentiment analysis using machine learning techniques on python," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2018, pp. 779–785.
- [36] P. Bo and L. Lee, "Opinion mining and sentiment analysis foundations and trends in information retrieval," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, p. 1135, 2008.
- [37] B. Gupta *et al.*, "Study of twitter sentiment analysis using machine learning algorithms on python," *International Journal of Computer Applications*, vol. 165, no. 9, pp. 29–34, 2017.
- [38] E. Hossain, O. Sharif, M. M. Hoque, and I. H. Sarker, "Sentilstm: A deep learning approach for sentiment analysis of restaurant reviews," in *International Conference on Hybrid Intelligent Systems*, Springer, 2020, pp. 193–203.

- [39] F. Sebastiani and A. Esuli, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of the 5th international conference on language resources and evaluation*, European Language Resources Association (ELRA) Genoa, Italy, 2006, pp. 417–422.
- [40] M. Thelwall, "The heart and soul of the web? sentiment strength detection in the social web with sentistrength," *Cyberemotions: Collective emotions in cyberspace*, pp. 119–134, 2017.
- [41] S. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, S. Manandhar and D. Yuret, Eds., Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 321–327. [Online]. Available: <https://aclanthology.org/S13-2053>.
- [42] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168–177.
- [43] M. M. Bradley and P. J. Lang, "Affective norms for english words (anew): Instruction manual and affective ratings," Technical report C-1, the center for research in psychophysiology ..., Tech. Rep., 1999.
- [44] V. Rentoumi, G. Giannakopoulos, V. Karkaletsis, and G. A. Vouros, "Sentiment analysis of figurative language using a word sense disambiguation approach," in *Proceedings of the International Conference RANLP-2009*, 2009, pp. 370–375.
- [45] A. Hogenboom, A. Brojba-Micu, and F. Frasincar, "The impact of word sense disambiguation on stock price prediction," *Expert systems with applications*, vol. 184, p. 115 568, 2021.
- [46] C. Strapparava and R. Mihalcea, "SemEval-2007 task 14: Affective text," in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, E. Agirre, L. Màrquez, and R. Wicentowski, Eds., Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 70–74. [Online]. Available: <https://aclanthology.org/S07-1013>.
- [47] R. L. Boyd, A. Ashokkumar, S. Seraj, and J. W. Pennebaker, "The development and psychometric properties of liwc-22," *Austin, TX: University of Texas at Austin*, vol. 10, 2022.
- [48] A. Koutsoumpis *et al.*, "The kernel of truth in text-based personality assessment: A meta-analysis of the relations between the big five and the linguistic inquiry and word count (liwc).," *Psychological Bulletin*, vol. 148, no. 11-12, p. 843, 2022.
- [49] A. Onan, "Sentiment analysis on twitter based on ensemble of psychological and linguistic feature sets," *Balkan Journal of Electrical and Computer Engineering*, vol. 6, no. 2, pp. 69–77, 2018.

- [50] C. Lee, K. Kim, J. Lim, and Y. Lee, "Psychological research using linguistic inquiry and word count (liwc) and korean linguistic inquiry and word count (kliwc) language analysis methodologies," *Journal of Cognitive Science*, vol. 16, no. 2, pp. 132–49, 2015.
- [51] Ł. Augustyniak, P. Szymański, T. Kajdanowicz, and W. TuligŁowicz, "Comprehensive study on lexicon-based ensemble classification sentiment analysis," *Entropy*, vol. 18, no. 1, 2016, ISSN: 1099-4300. DOI: [10.3390/e18010004](https://doi.org/10.3390/e18010004). [Online]. Available: <https://www.mdpi.com/1099-4300/18/1/4>.
- [52] Y. S. Mehanna and M. B. Mahmuddin, "A semantic conceptualization using tagged bag-of-concepts for sentiment analysis," *IEEE Access*, vol. 9, pp. 118 736–118 756, 2021.
- [53] S.-A. Bahrainian and A. Dengel, "Sentiment analysis using sentiment features," in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, IEEE, vol. 3, 2013, pp. 26–29.
- [54] A. Alrehili and K. Albalawi, "Sentiment analysis of customer reviews using ensemble method," in *2019 International conference on computer and information sciences (ICCIS)*, IEEE, 2019, pp. 1–6.
- [55] B. S. Rintyarna, R. Sarno, and C. Fatichah, "Evaluating the performance of sentence level features and domain sensitive features of product reviews on supervised sentiment analysis tasks," *Journal of Big Data*, vol. 6, no. 1, pp. 1–19, 2019.
- [56] M. Alshammari and M. Mezher, "A comparative analysis of data mining techniques on breast cancer diagnosis data using weka toolbox," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 8, pp. 224–229, 2020.
- [57] S. N. Almuayqil *et al.*, "Framework for improved sentiment analysis via random minority oversampling for user tweet review classification," *Electronics*, vol. 11, no. 19, 2022, ISSN: 2079-9292. DOI: [10.3390/electronics11193058](https://doi.org/10.3390/electronics11193058). [Online]. Available: <https://www.mdpi.com/2079-9292/11/19/3058>.
- [58] M. V. Valero, "Thousands of scientists are cutting back on twitter," *Nature*, vol. 620, pp. 482–4, 2023.
- [59] S. Tripathi, R. Mehrotra, V. Bansal, and S. Upadhyay, "Analyzing sentiment using imdb dataset," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, IEEE, 2020, pp. 30–33.
- [60] K. Venkata Raju and M. Sridhar, "Based sentiment prediction of rating using natural language processing sentence-level sentiment analysis with bag-of-words approach," in *First International Conference on Sustainable Technologies for Computational Intelligence: Proceedings of ICTSCI 2019*, Springer, 2020, pp. 807–821.

- [61] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335–4385, 2020.
- [62] A. Aldoseri, K. N. Al-Khalifa, and A. M. Hamouda, "Re-thinking data strategy and integration for artificial intelligence: Concepts, opportunities, and challenges," *Applied Sciences*, vol. 13, no. 12, p. 7082, 2023.
- [63] B. van Giffen, D. Herhausen, and T. Fahse, "Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods," *Journal of Business Research*, vol. 144, pp. 93–106, 2022, ISSN: 0148-2963. DOI: <https://doi.org/10.1016/j.jbusres.2022.01.076>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0148296322000881>.
- [64] H. Pouransari and S. Ghili, "Deep learning for sentiment analysis of movie reviews," *CS224N Proj*, <https://cs224d.stanford.edu/reports/PouransariHadi.pdf>, vol. 0, pp. 1–8, 2014.
- [65] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [66] G. Xu *et al.*, "Sentiment analysis of comment texts based on bilstm," *Ieee Access*, vol. 7, pp. 51 522–51 532, 2019.
- [67] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," *Information*, vol. 15, no. 9, p. 517, 2024.
- [68] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [69] C. Eang and S. Lee, "Improving the accuracy and effectiveness of text classification based on the integration of the bert model and a recurrent neural network (rnn_{bert_based})," *Applied Sciences*, vol. 14, no. 18, 2024, ISSN: 2076-3417. DOI: [10.3390/app14188388](https://doi.org/10.3390/app14188388). [Online]. Available: <https://www.mdpi.com/2076-3417/14/18/8388>.
- [70] T. B. Brown, "Language models are few-shot learners," *arXiv preprint ArXiv:2005.14165*, 2020.
- [71] F. Kamalov, D. Santandreu Calonge, and I. Gurrib, "New era of artificial intelligence in education: Towards a sustainable multifaceted revolution," *Sustainability*, vol. 15, no. 16, 2023, ISSN: 2071-1050. DOI: [10.3390/su151612451](https://doi.org/10.3390/su151612451). [Online]. Available: <https://www.mdpi.com/2071-1050/15/16/12451>.
- [72] M. A. Al-Sharafi *et al.*, "Understanding the impact of knowledge management factors on the sustainable use of ai-based chatbots for educational purposes using a hybrid sem-ann approach," *Interactive Learning Environments*, pp. 1–20, 2022.
- [73] K. I. Roumeliotis and N. D. Tselikas, "Chatgpt and open-ai models: A preliminary review," *Future Internet*, vol. 15, no. 6, p. 192, 2023.

- [74] T. TALAN and Y. KALINKARA, "The role of artificial intelligence in higher education: Chatgpt assessment for anatomy course," *Uluslararası Yönetim Bil-işim Sistemleri ve Bilgisayar Bilimleri Dergisi*, vol. 7, no. 1, pp. 33–40, 2023.
- [75] J. G. Meyer *et al.*, "Chatgpt and large language models in academia: Opportunities and challenges," *BioData Mining*, vol. 16, no. 1, p. 20, 2023.
- [76] P. P. Ray, "Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, 2023.
- [77] A. Korkmaz, C. Aktürk, and T. TALAN, "Analyzing the user's sentiments of chatgpt using twitter data," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 2, pp. 202–214, 2023.
- [78] C. A. Gao *et al.*, "Comparing scientific abstracts generated by chatgpt to real abstracts with detectors and blinded human reviewers," *NPJ Digital Medicine*, vol. 6, no. 1, p. 75, 2023.
- [79] F. Sudirjo *et al.*, "Application of chatgpt in improving customer sentiment analysis for businesses," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 3, pp. 283–288, 2023.
- [80] D. Araci, *Finbert: Financial sentiment analysis with pre-trained language models*, 2019. [Online]. Available: <https://arxiv.org/abs/1908.10063>.
- [81] Y. Yang, M. C. S. Uy, and A. Huang, "Finbert: A pretrained language model for financial communications," *arXiv preprint arXiv:2006.08097*, 2020.
- [82] X.-Y. Liu, G. Wang, H. Yang, and D. Zha, "Fingpt: Democratizing internet-scale data for financial large language models," *arXiv preprint arXiv:2307.10485*, 2023.
- [83] S. Gite *et al.*, "Explainable stock prices prediction from financial news articles using sentiment analysis," *PeerJ Computer Science*, vol. 7, e340, 2021.
- [84] A. Sharaff, T. R. Chowdhury, and S. Bhandarkar, "Lstm based sentiment analysis of financial news," *SN Computer Science*, vol. 4, no. 5, p. 584, 2023.
- [85] K. Mishev *et al.*, "Evaluation of sentiment analysis in finance: From lexicons to transformers," *IEEE Access*, vol. 8, pp. 131 662–131 682, 2020, ISSN: 2169-3536. DOI: [10.1109/ACCESS.2020.3009626](https://doi.org/10.1109/ACCESS.2020.3009626).
- [86] A. M. Ardekani *et al.*, "Finsentgpt: A universal financial sentiment engine?" *International Review of Financial Analysis*, vol. 94, p. 103 291, 2024, ISSN: 1057-5219. DOI: <https://doi.org/10.1016/j.irfa.2024.103291>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1057521924002230>.
- [87] T. Joachims, "A support vector method for multivariate performance measures," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 377–384.
- [88] Ragas, *Metrics*, <https://docs.ragas.io/en/latest/concepts/metrics/index.html>, 2024.

- [89] M. S. Akhtar *et al.*, "All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework," *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 285–297, 2022. DOI: [10.1109/TAFFC.2019.2926724](https://doi.org/10.1109/TAFFC.2019.2926724).
- [90] L.-C. Yu, J.-L. Wu, P.-C. Chang, and H.-S. Chu, "Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news," *Knowledge-Based Systems*, vol. 41, pp. 89–97, 2013, ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knosys.2013.01.001>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S095070511300004X>.
- [91] N. Jing, Z. Wu, and H. Wang, "A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction," *Expert Systems with Applications*, vol. 178, p. 115 019, 2021.
- [92] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *arXiv: Computation and Language*, 2020. DOI: [null](#).
- [93] M. Rehan, M. S. I. Malik, and M. M. Jamjoom, "Fine-tuning transformer models using transfer learning for multilingual threatening text identification," *IEEE Access*, 2023.
- [94] Z. Du *et al.*, "All nlp tasks are generation tasks: A general pretraining framework," *arXiv.org*, 2021. DOI: [null](#).
- [95] Y. Tao, B. Godefroy, G. Genthial, and C. Potts, "Effective feature representation for clinical text concept extraction," in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, A. Rumshisky, K. Roberts, S. Bethard, and T. Naumann, Eds., Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 1–14. DOI: [10.18653/v1/W19-1901](https://doi.org/10.18653/v1/W19-1901). [Online]. Available: <https://aclanthology.org/W19-1901>.
- [96] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [97] V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.
- [98] I. Vulic *et al.*, "Convfit: Conversational fine-tuning of pretrained language models," *Conference on Empirical Methods in Natural Language Processing*, 2021. DOI: [10.18653/v1/2021.emnlp-main.88](https://doi.org/10.18653/v1/2021.emnlp-main.88).
- [99] J. Lee *et al.*, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.
- [100] R. Obiedat *et al.*, "Sentiment analysis of customers' reviews using a hybrid evolutionary svm-based approach in an imbalanced data distribution," *IEEE Access*, vol. 10, pp. 22 260–22 273, 2022.
- [101] E. Fersini, E. Messina, and F. A. Pozzi, "Sentiment analysis: Bayesian ensemble learning," *Decision support systems*, vol. 68, pp. 26–38, 2014.

- [102] A. A. Altowayan and A. Elnagar, "Improving arabic sentiment analysis with sentiment-specific embeddings," in *2017 IEEE international conference on big data (big data)*, IEEE, 2017, pp. 4314–4320.
- [103] B. O. Deho, A. W. Agangiba, L. F. Aryeh, and A. J. Ansah, "Sentiment analysis with word embedding," in *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*, IEEE, 2018, pp. 1–4.
- [104] A. Pathak *et al.*, "Performance analysis of lora finetuning llama-2," in *2023 7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, IEEE, 2023, pp. 1–4.
- [105] Q. Long, W. Wang, and S. Pan, "Adapt in contexts: Retrieval-augmented domain adaptation via in-context learning," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 6525–6542.
- [106] S. Wu *et al.*, *Retrieval-augmented generation for natural language processing: A survey*, 2024. [Online]. Available: <https://arxiv.org/abs/2407.13193>.
- [107] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," *Conference of the European Chapter of the Association for Computational Linguistics*, 2023. DOI: [10.48550/arxiv.2309.15217](https://doi.org/10.48550/arxiv.2309.15217).
- [108] L. Yang *et al.*, "Trustworthy llms: A survey and guideline for evaluating large language models' alignment," *arXiv.org*, 2023. DOI: [10.48550/arxiv.2308.05374](https://doi.org/10.48550/arxiv.2308.05374).
- [109] B. Zhang *et al.*, "Enhancing financial sentiment analysis via retrieval augmented large language models," in *Proceedings of the fourth ACM international conference on AI in finance*, 2023, pp. 349–356.
- [110] G. Bathla, R. Rani, and H. Aggarwal, "Stocks of year 2020: Prediction of high variations in stock prices using lstm," *Multimedia Tools and Applications*, vol. 82, no. 7, pp. 9727–9743, 2023.
- [111] F. N. Khan, T. Asim, S. M. Saleem, and R. A. Rana, "Finchat: Knowledge-driven financial expert," Ph.D. dissertation, Habib University, 2022.
- [112] Absa, "Absa investor relations," *Website*, <https://www.absa.africa/investor-relations/financial-results/>, 2023, Accessed on August 30, 2023.
- [113] Capitec Bank, "Capitec bank financial results 2023," *Website*, <https://www.capitecbank.co.za/financial-results/2023/>, 2023, Accessed on September 5, 2023.
- [114] Nedbank, "Nedbank investor relations," *Website*, <https://www.nedbank.co.za/content/nedbank/desktop/governance/investor-relations/information-hub/financial-results.html>, 2023, Accessed on August 10, 2023.
- [115] Standard Bank, "Standard bank financial results," *Website*, <https://reporting.standardbank.com/results-reports/financial-results/>, 2023, Accessed on July 22, 2023.
- [116] FirstRand, "FirstRand integrated reporting hub," *Website*, <https://www.firstrand.co.za/investors/integrated-reporting-hub/financial-reporting/>, 2023, Accessed on July 15, 2023.

- [117] D. W. Arner, J. Barberis, and R. P. Buckley, "Fintech, regtech, and the reconceptualization of financial regulation," *Nw. J. Int'l L. & Bus.*, vol. 37, p. 371, 2016.
- [118] Z. K. Chomba, *The Impact of the COVID-19 pandemic on the South African Equity Market: An Event Study Analysis*. University of Johannesburg (South Africa), 2021.
- [119] R. Harrisberg, "An analysis of the low-volatility anomaly on the johannesburg stock exchange," M.S. thesis, Faculty of Commerce ,Department of Finance and Tax, University of Cape Town, South Africa, 2019. [Online]. Available: <https://open.uct.ac.za/items/b2758f2c-3339-4dcc-a6fd-21624599cb00>.
- [120] D. Tang, B. Qin, and T. Liu, *Aspect level sentiment classification with deep memory network*, 2016. [Online]. Available: <https://arxiv.org/abs/1605.08900>.
- [121] A. Maas *et al.*, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.
- [122] S. Bagherzadeh, S. Shokouhyar, H. Jahani, and M. Sigala, "A generalizable sentiment analysis method for creating a hotel dictionary: Using big data on tripadvisor hotel reviews," *Journal of Hospitality and Tourism Technology*, vol. 12, no. 2, pp. 210–238, 2021.
- [123] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, "Stance and sentiment in tweets," *ACM Transactions on Internet Technology (TOIT)*, vol. 17, no. 3, pp. 1–23, 2017.
- [124] P. Malo *et al.*, "Good debt or bad debt: Detecting semantic orientations in economic texts," *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, pp. 782–796, 2014.
- [125] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," *International journal of machine learning and cybernetics*, vol. 1, pp. 43–52, 2010.
- [126] W. B. Cavnar, J. M. Trenkle, *et al.*, "N-gram-based text categorization," in *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, Ann Arbor, Michigan, vol. 161175, 1994, p. 14.
- [127] A. Aizawa, "An information-theoretic perspective of tf-idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.
- [128] M. Roederer *et al.*, "Probability binning comparison: A metric for quantitating multivariate distribution differences," *Cytometry: The Journal of the International Society for Analytical Cytology*, vol. 45, no. 1, pp. 47–55, 2001.
- [129] J. S. Kessler and C. Global, "Scattertext: A browser-based tool for visualizing how corpora differ," *ACL 2017*, p. 85, 2017.
- [130] F. Nausheen and S. H. Begum, "Sentiment analysis to predict election results using python," in *2018 2nd international conference on inventive systems and control (ICISC)*, IEEE, 2018, pp. 1259–1262.

- [131] C. Kaur and A. Sharma, "Social issues sentiment analysis using python," in *2020 5th international conference on computing, communication and security (ICCCS)*, IEEE, 2020, pp. 1–6.
- [132] Y. Heryanto and A. Triayudi, "Evaluating text quality of gpt engine davinci-003 and gpt engine davinci generation using bleu score," *SAGA: Journal of Technology and Information System*, vol. 1, no. 4, pp. 121–129, 2023.
- [133] J. Nay, "Large language models as corporate lobbyists," *arXiv preprint arXiv:2301.01181*, 2023.
- [134] Z. Nanli, Z. Ping, L. Weiguo, and C. Meng, "Sentiment analysis: A literature review," in *2012 International Symposium on Management of Technology (ISMOT)*, IEEE, 2012, pp. 572–576.
- [135] S. Ahuja and G. Dubey, "Clustering and sentiment analysis on twitter data," in *2017 2nd International Conference on Telecommunication and Networks (TELNET)*, IEEE, 2017, pp. 1–5.
- [136] I. G. S. M. Diyasa *et al.*, "Twitter sentiment analysis as an evaluation and service base on python textblob," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 1125, 2021, p. 012 034. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1757-899X/1125/1/012034/pdf>.
- [137] S. Elbagir and J. Yang, "Sentiment analysis on twitter with python's natural language toolkit and vader sentiment analyzer," in *IAENG Transactions on Engineering Sciences: Special Issue for the International Association of Engineers Conferences 2019*, World Scientific, 2020, pp. 63–80.
- [138] I. Hemalatha, G. S. Varma, and A. Govardhan, "Preprocessing the informal text for efficient sentiment analysis," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 1, no. 2, pp. 58–61, 2012.
- [139] S. Elbagir and J. Yang, "Twitter sentiment analysis using natural language toolkit and vader sentiment," in *Proceedings of the international multiconference of engineers and computer scientists*, vol. 122, 2019, p. 16.
- [140] A. Borg and M. Boldt, "Using vader sentiment and svm for predicting customer response sentiment," *Expert Systems with Applications*, vol. 162, p. 113 746, 2020.
- [141] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
- [142] S. Bates, T. Hastie, and R. Tibshirani, "Cross-validation: What does it estimate and how well does it do it?" *Journal of the American Statistical Association*, pp. 1–12, 2023.
- [143] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.
- [144] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD*

- international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.
- [145] S. Gururangan *et al.*, “Don’t stop pretraining: Adapt language models to domains and tasks,” *arXiv preprint arXiv:2004.10964*, 2020.
 - [146] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, “Transfer learning in natural language processing,” in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, 2019, pp. 15–18.
 - [147] J. Gao, M. Galley, and L. Li. 2019. DOI: [10.1561/15000000074](https://doi.org/10.1561/15000000074).
 - [148] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.
 - [149] J. Wei *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
 - [150] S. Yao *et al.*, “React: Synergizing reasoning and acting in language models,” *arXiv preprint arXiv:2210.03629*, 2022.
 - [151] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *arXiv preprint arXiv:1908.10084*, 2019.
 - [152] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,” *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.
 - [153] H. Touvron *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
 - [154] J. Achiam *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
 - [155] S. Coyne *et al.*, “Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction,” *arXiv preprint arXiv:2303.14342*, 2023.
 - [156] A. Dubey *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.
 - [157] E. Strubell, A. Ganesh, and A. McCallum, “Energy and policy considerations for modern deep learning research,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 13 693–13 696.
 - [158] H. Jegou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.
 - [159] R. Guo *et al.*, “Accelerating large-scale inference with anisotropic vector quantization,” in *International Conference on Machine Learning*, PMLR, 2020, pp. 3887–3896.
 - [160] W. Chen *et al.*, “Hierarchical quantization for billion-scale similarity retrieval on gpus,” *Computers & Electrical Engineering*, vol. 90, p. 107 002, 2021.
 - [161] S. Lermen, C. Rogers-Smith, and J. Ladish, “Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b,” *arXiv preprint arXiv:2310.20624*, 2023.

- [162] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale," *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 318–30 332, 2022.
- [163] E. J. Hu *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.
- [164] E. Perez, D. Kiela, and K. Cho, "True few-shot learning with language models," *Advances in neural information processing systems*, vol. 34, pp. 11 054–11 070, 2021.
- [165] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.
- [166] S. Abimannan *et al.*, "Ensemble multifeatured deep learning models and applications: A survey," *IEEE Access*, 2023.
- [167] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.
- [168] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, Springer, 2000, pp. 1–15.
- [169] L. I. Kuncheva, *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.
- [170] T.-Y. Lin *et al.*, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [171] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural networks*, vol. 106, pp. 249–259, 2018.
- [172] A. P. Pasarkar and A. B. Dieng, "Cousins of the vendi score: A family of similarity-based diversity metrics for science and machine learning," *arXiv preprint arXiv:2310.12952*, 2023.
- [173] D. Friedman and A. B. Dieng, "The vendi score: A diversity evaluation metric for machine learning," *arXiv preprint arXiv:2210.02410*, 2022.