BI-LSTM BASED WEAKLY SUPERVENED ENHANCING EMOTIONAL SENTIMENT ANALYSIS IN E-COMMERCE REVIEWS

A PROJECT REPORT

Submitted by

MANIMALAR S

(2023176042))

A report for the phase-I of the project
submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment
for the award of the degree
of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

SPECIALIZATION IN AI & DS

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600 025

JAN 2025

ANNA UNIVERSITY

CHENNAI - 600 025

BONA FIDE CERTIFICATE

Certified that this project report titled BI-LSTM BASED WEAKLY SUPERVENED ENHANCING EMOTIONAL SENTIMENT ANALYSIS IN E-COMMERCE REVIEWS is the bona fide work of MANIMALAR S(2023176042) who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:CHENNAI
DATE:

< DR S SRIDHAR>
< PROFESSOR>
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

DR. S. SWAMYNATHAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

Always there is a demand of ABSC for product review as it improves the business by means of decision making process, customer experience and a marketing strategy efficiently. Many model are available for emotion (aspect) based sentiment classification using deep learning , self learning ,reinforcement learning etc.

A model is build to understand sentiment with respect to aspects like price, service, quality, size and usability automatically. Three sets of amazon product review data are chosen whose columns are reviews-text and ratings for training, testing and validation. A novel approach to sentiment classification based on aspect category along with aspect terms in review using Bi-LSTM ,DeBERT and HABSC algorithm.

The aim is to classify the sentiment for unlabeled data by extracting the aspect term present inthe customer review under the aspect category such as price, size, quality, usability and service with use of model.BilSTM model is prioritized as it facilitate semantic comprehensive by extract context from past and future states. Advantages of BiLSTM are more important to understand emotion(aspect) present in reviews. At first data are labeled according to the aspect term using the auto-label algorithm. Using snorkel sentiment classification matches manual sentiment classification by 97 percent which is set as the target or true label. As of all Bi-LSTM does sentiment classifies best. This method offers more accurate aspect-based sentiment analysis than conventional methods, providing e-commerce platforms with insightful data on user input which helps to be specifically in recommendation system and for better results try more innovative method.

Keywords: Bi-LSTM (Bidirectional Long Short-Term Memory), ABSC(Aspect Based Sentiment Classification), HABSC(Hybrid Aspect Based Sentiment Classification), Snorkel framework.

ACKNOWLEDGEMENT

It is my privilege to express my deepest sense of gratitude and sincere thanks to Dr.S.SRIDHAR, Professor, Department of Information Science and Technology, College of Engineering , Guindy , Anna University , for her constant supervision, encouragement, and support in my project work.

I greatly appreciate the constructive advice and motivation that was given to help me advance my project in the right direction. I am grateful to Dr. S. SWAMYNATHAN,HOD,Department of Information Science and Technology, College of Engineering Guindy , Anna University for providing us with the opportunity and necessary resources to do this project.

I would also wish to express my deepest sense of gratitude to the Members of the Project Review Committee : Dr . S.SRIDHAR , Professor, Dr. G. GEETHA, Associate Professor , Dr . D . NARASHIMAN,Teaching Fellow Department of Information Science and Technology , College of Engineering Guindy, Anna University, for their guidance and useful suggestions that were beneficial in helping me improve my project.

I also thank the faculty member and non teaching staff members of the Department of Information Science and Technology, Anna University, Chennai for their valuable support throughout the course of our project work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

$-$, $\neg$, $\sim$	Negation operator
$*$, $\wedge$	Multiplication operator
$+$, $\vee$, $\cup$	Disjunction operator
X, $\wedge$	Conjunction operator
$\rightarrow$	Conditional operator
$\leftrightarrow$	Biconditional operator
$\diamond$	Future tense modal operator
α	Action

CHAPTER 1

INTRODUCTION

As of today's market customer reviews have become an invaluable source of information for both businesses and consumers.There is an overwhelming volume of product reviews which helps to gain insights into user preferences, identify areas of improvement, and enhance overall customer satisfaction.As large volume of reviews makes manual analysis impractical leads to necessity of automated sentiment analysis solutions. Aspect-Based Sentiment Classification (ABSC) offer granular approach to understanding customer sentiment. Aspect-based sentiment classification (ABSC) plays a crucial role in this process by analyzing customer reviews to determine sentiments associated with specific aspects such as price, quality, service, size, and usability will enhance informed decisions about product improvements, marketing strategies, and customer service enhancements. This research introduces a novel approach for a comprehensive and automated ABSC model using a combination of advanced deep learning techniques, specifically Bi-LSTM (Bidirectional Long Short-Term Memory), DeBERT, and HABSC algorithms.The proposed system addresses accurate aspect term extraction from unstructured review text,complexity of understanding context-dependent sentiments,requirement of automated 2 processing unlabeled data,need of high accuracy across different aspect categories.

1.1 PROBLEM STATEMENT

At present challenge in e-commerce product review analysis lies in accurately extracting and classifying sentiments associated with specific aspects of products (such as price, service, quality, size, and usability) from unstructured customer reviews. Traditional sentiment analysis methods often fail to capture the nuanced relationships between different aspects and their associated sentiments, leading to incomplete or inaccurate understanding of customer feedback.Some of problem statement are listed below. [**?**].

- The need to automatic identification and extract aspect terms from unlabeled customer review data.
- Better identification of product advantages and disadvantages based on specific aspects
- Improved decision-making aspect-specific feedback.
- The necessity to handle complex semantic relationships and context in review text.
- The challenge of creating a reliable automated labeling system for training data.
- Accurate extraction and classification of sentiment for specific aspect terms embedded in customer reviews.
- The demand for high accuracy in sentiment classification to support business decision-making.

1.2 OBJECTIVE

Project aim is to predict sentiment classification with respect to aspect term and aspect category for unlabeled data so that it can avoid manual annotation for labeling which reduce time and effort during sentimental analysis insightful. To build this model, other supporting objectives are automate labeling, aspect-based sentiment classification, leverage weak supervision, enhance generalization for all types of product and ensure scalability.

- Aim is to accurately classify sentiments with respect to various aspect categories and its respective terms in customer reviews, ensuring better insights for e-commerce platforms
- To develop an advanced Aspect-Based Sentiment Classification (ABSC) model using Bi-LSTM, DeBERT, and HABSC algorithms to automate the extraction and classification of aspect terms from unlabeled customer reviews
- Achieve high accuracy in aspect based sentiment classification.
- Construct a more effective mechanism to handle unlabeled review data

1.3 OVERVIEW

This project aims to revolutionize sentiment analysis in e-commerce by addressing the challenges of large-scale data annotation in base of aspect. The system processes Amazon product reviews to classify sentiments across five key aspects: price, size, quality, usability, and service. A unique technique that automatically labels unlabeled e-commerce review data is implemented at initial stage of the project to minimize the requirement for manual annotations. Using snorkel sentiment label is done in later part which is used as true label. A customized multi model sentiment classification method with ensemble based decision making is then used to categorize sentiments unique to aspect terms whose inputs are ratings, reviews and aspect keyword which is an external resources that achieves of about 85 percent accuracy. A hybrid BiLSTM Model with ratings ,reviews and along with their aspect terms are defined by means of concatenation and dense layer predicted ABSC. Likewise hybrid DeBERT model uses self-attention strategy to concentrate on aspect phrases, which improves the accuracy of sentiment classification. The hybrid model is thus generalize across various product categories and review types, ensuring robustness and scalability to process large volumes of review data. By combining automated labeling, aspect-specific analysis, weak supervision, and self-attention, this project aims to deliver an efficient, scalable, and accurate sentiment analysis model for e-commerce platforms.

1.4 SCOPE OF THE PROJECT

The proposed research introduces an advanced framework for sentiment analysis focusing on creating an autonomous, sophisticated algorithmic approach that minimizes human intervention, leverages implicit rating information, and develops intelligent neural network architectures.

- Processing and analysis of unlabeled Amazon product review datasets
- Implement a Snorkel-based auto-labeling approach to create labeled datasets efficiently, ensuring high accuracy and consistency.
- Develop an algorithm to extract aspect terms automatically and associate them with predefined categories (e.g., price, quality, usability, size and services).
- Implementation of Hybrid Bi-LSTM, DeBERT models, and HABSC algorithm for comparative analysis.
- Sentiment classification for five specific aspect categories as price, size, service, usability and quality.
- Compare the proposed models(Bi-LSTM,DeBERT,HABSC) based on metrics like precision, recall, F1-score, and hamming loss, highlighting Bi-LSTM's superior performance in aspect-based sentiment classification.

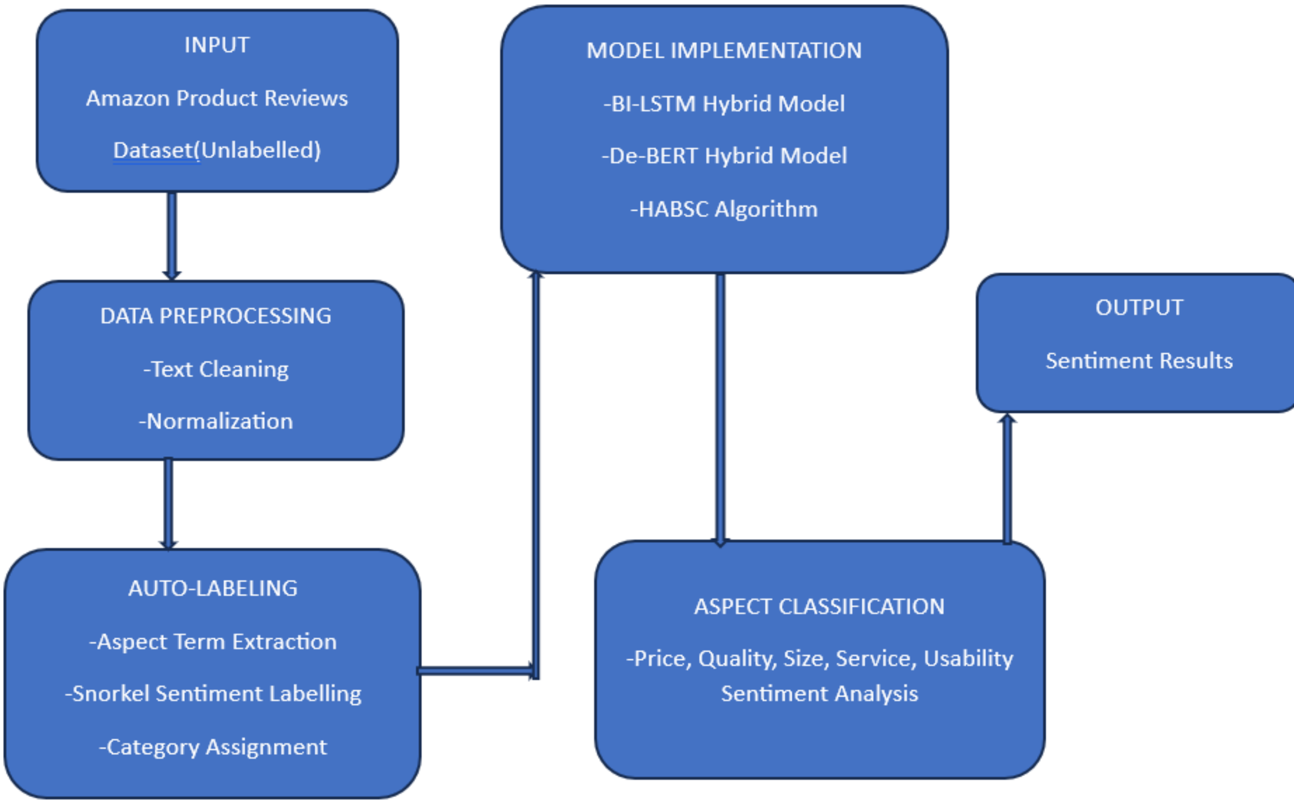

Figure 1.1: BLOCK DIAGRAM

1.5 MODULES

A project is divide into discrete units of functionality and explained each and every modules along with its functionality in details as below.

1.5.1 Data Pre-processing Module

Data is collected from resources present in two famous website in form of unlabeled text. Three sets of dataset are used as combine. Nature of dataset has attributes as ratings and reviews.The data is processed by removing links (noise) ,handle missing values and normalized text.Text is cleaned , tokenization ,lemmatization and vectorization such that it suits for embedding.

1.5.2 Weak Signal Generation

Generate label for the each and every reviews by auto-labeling algorithm called keyword extraction algorithm.Among all aspect important aspect as "price, usability, service, size and quality" are chosen for aspect based sentiment analysis.Open source snorkel framework is used for sentiment label which is later used as target label.Aspect terms and aspect category are labeled using external resource(aspect dictionary).

1.5.3 Build Model and Training

HABSC is heuristic based which is combined weak score of sentiwordnet score, vadar score and bert score to predict sentiment classification as neutral ,positive and negative based on condition. BiLSTMhybrid model which consist of aspect terms ,ratings and aspect category as weak signal input which is concated to predict sentiment classification. DeBERT hybrid model has weeak signals to predict sentiment classification.

1.5.4 Evaluation

After training the model and algorithm is being evaluated by means of metrics like accuracy, precision, recall,Hamming loss ,macro precision , micro precision, macro recall , micro recall , micro f1-score , macro f1 score and F1 score to ensure effective emotional prediction with e-commerce perspective.

CHAPTER 2

LITERATURE SURVEY/RELATED WORK

A literature review provides a crucial basis for any research project by examining current approaches, paradigms, and issues in the f ield, providing insightful information on the state of the art. It aids in locating research gaps, revealing topics that need more investigation, and providing context for the current work by placing it within the larger f ield. Through examining previous research, scientists might find ideas for creating or refining procedures while making sure their work adds new insights rather than repeating previously conducted studies. This procedure not only supports the importance of the study but also directs the choice of practical methods and strategies, opening the door for significant breakthroughs in the area.

2.1 INTRODUCTION

A subfield of sentiment analysis called Aspect-Based Sentiment Analysis (ABSA) is dedicated to identifying and categorizing sentiments at the aspect level. By concentrating on particular facets or characteristics of entities (such as goods, services, or experiences), this method seeks to examine the subjective view points presented in reviews, social media posts, and other text sources. In many applications, including social media analytics, brand reputation management, and 10 market research, ABSA is essential. By concentrating on particular factors like cost, quality, usability, and customer service, ABSA enables a more detailed analysis than standard sentiment analysis, which groups sentiments according to the general text polarity (positive, negative, or neutral).

In recent years, ABSA has evolved with the advent of more sophisticated methods, particularly deep learning approaches, to improve accuracy and address complex challenges such as aspect extraction, aspect sentiment classification, and sentiment evolution over time. This paper reviews the state-of-the-art approaches in ABSA, highlighting the strengths and limitations of existing systems, and proposes future directions to enhance performance across diverse domains and languages.

2.2 EXISTING SYSTEM

ABSA methods can be broadly categorized into three main approaches: lexicon-based methods, traditional machine learning methods, and deep learning methods.

- **Lexicon-Based Methods** Lexicon-based approaches use pre-made lexicons—dictionaries of words that convey sentiment—to link certain elements to sentiment polarity. These techniques function by locating keywords associated with a certain topic in a text and using the lexicon to 11 infer sentiment polarity. These approaches, however, are constrained by their dependence on manually created lexicons and find it difficult to deal with ambiguous or domain-specific language.
- **Traditional Machine Learning Methods:** ABSA is handled as a multi-class classification problem in traditional machine learning techniques. Aspect sentiment is classified using well-known methods including Support Vector Machines (SVM), Decision Trees, and Naive Bayes. Relevant characteristics must be manually extracted from the text and feature engineering is required for these models. Even though they produce respectable results, they struggle to handle complicated connections between attributes and attitudes and

enormous amounts of data.

- **Deep Learning Methods** ABSA has been transformed by deep learning algorithms, which automate feature extraction and create intricate correlations between attitudes and attributes. Attention-based processes like Transformers (e.g., BERT) and models like Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been used to enhance ABSA performance. These models can handle massive amounts of unstructured text, learn rich representations from data, and provide higher accuracy than conventional techniques. Nonetheless, 12 issues including implicit sentiment expression, multilingual sentiment analysis, and domain adaptability continue to exist.

2.3 PROPOSED SYSTEM

Recent advancements in ABSA have focused on addressing several challenges and improving the robustness of sentiment classification

- **Domain-Dependent Study:** One of the critical challenges in ABSA is domain dependence. Models trained on one domain (e.g., electronics) may perform poorly in another (e.g., movies) due to differences in sentiment word meanings. Researchers have proposed methods for transferring sentiment knowledge across domains, using sentiment graphs and domain-specific feature extraction techniques.
- **Multi-modal Analysis:** Future ABSA systems should include more data types, such pictures, videos, and emoticons, to improve sentiment categorization in light of the growing amount of multi-modal data on social media sites. By using this method, ABSA

systems would be able to use more comprehensive information sources than only text.

- **Multilingual Sentiment Analysis:** The creation of ABSA systems that can manage low-resource languages like Hindi, Telugu, and others is essential because social media and online 13 reviews are frequently multilingual. Promising methods for enhancing multilingual ABSA systems include cross-lingual transfer learning techniques and trained language models.

- **BiLSTM in ABSA:**BiLSTMmodelsprovidearobustsolution to ABSA challenges by leveraging bidirectional sequence processing to capture contextual nuances in sentiment analysis. Their ability to process sequences in both directions enables accurate contextual understanding of aspects within sentences, while also handling complex relationships between different aspects and their associated sentiments through consideration of both past and future contexts. BiLSTMs excel at tracking sentiment evolution over time, making them valuable for analyzing long-term datasets like product reviews or social media posts, and their adaptability to different languages and domains allows them to address domain-specific sentiment variations through fine-tuning on specialized datasets, ultimately improving sentiment analysis accuracy across diverse applications.

2.4 CONCLUSION

The advent of deep learning methods, particularly pre-trained language models like BERT, has greatly advanced the field of aspect-based sentiment analysis. By addressing complicated relationships in text, 14 increasing classification accuracy, and automating feature extraction, these models have raised the bar for ABSA. Nonetheless, a number of issues have to be resolved, such as implicit sentiment expression, multilingual analysis, and domain adaptability.

Future studies should concentrate on creating systems that can integrate multi-modal data, adapt across domains, and handle a variety of languages.Moreover, the need for more effective data pre-processing techniques, as well as better integration of commonsense knowledge, remains critical to enhancing the performance of ABSA systems. By addressing these challenges, the next generation of ABSA systems will be better equipped to provide nuanced and actionable insights into customer sentiments across various industries and languages.

In conclusion, BiLSTM has proven itself as a trans-formative tool in Aspect-based Sentiment Analysis, distinguishing itself from traditional methods through its sophisticated ability to process contextual relationships and complex sentence structures. This capability is particularly valuable for handling multifaceted challenges like multiple aspects, implicit sentiment expression, and temporal sentiment evolution in real-world applications such as customer feedback analysis and social media sentiment tracking. The model's bidirectional processing architecture enables superior accuracy in sentiment analysis, delivering detailed aspect-level insights that empower business decision-making. Looking ahead, the continued integration of BiLSTM with cutting-edge deep learning techniques In conclusion, BiLSTM has proven itself as a trans-formative tool in Aspect-based Sentiment Analysis, distinguishing itself from traditional methods through its sophisticated ability to process contextual relationships and complex sentence structures. This capability is particularly valuable for handling multifaceted challenges like multiple aspects, implicit sentiment expression, and temporal sentiment evolution in real-world applications such as customer feedback analysis and social media sentiment tracking. The model's bidirectional processing architecture enables superior accuracy in sentiment analysis, delivering detailed aspect-level insights that empower business decision-making. Looking ahead, the continued integration of BiLSTM with cutting-edge deep learning techniquesIn conclusion, BiLSTM has proven itself as a trans-formative tool in Aspect-based Sentiment Analysis, distinguishing itself from traditional methods through its sophisticated ability to process contextual relationships and complex sentence structures. This capability is particularly valuable for handling multifaceted challenges like multiple aspects, implicit sentiment expression, and temporal sentiment evolution in real-world applications such as customer feedback analysis and social media sentiment tracking. The model's bidirectional processing architecture enables superior accuracy in sentiment analysis, delivering detailed aspect-level insights that empower business decision-making. Looking ahead, the continued integration of BiLSTM with cutting-edge deep learning techniques.

CHAPTER 3

SYSTEM ARCHITECTURE

The architecture of a project integrates manual data editing, Snorkel for automatic sentiment labeling, and two hybrid models namely BiLSTM and DeBERT for aspect-based sentiment classification. The hybrid system architecture for sentiment classification undergoes the following process, as data collection, data pre-processing, data sampling building model, and model evaluation The goal is to compare Snorkel-labeled sentiment labels with predicted label from the hybrid model and HABSC algorithm.

3.1 System Architecture Description

As it is aspect based sentiment classification emotion to be considerate so lexicon based sentiment analysis plays a vital role as well adoptable one.The hybrid algorithm (HABSC) which predict emotional sentiment analysis where sentiment score which is combination of ensemble based decision making uses three types of score that ensure lexicon based sentiment analysis. .For hybrid model which used BERT and BILSTM is used These models are evaluated based on accuracy, precision, recall, and F1 score to identify the best model for deployment

3.2 Data Collection and Pre-processing

The system receives data from three different datasets:BM-amazon reviews , AR-amazon-product-reviews and amazon product review which are available in geeks-2-geeks and kaggle website .Initially only product review is filtered. Each dataset undergoes a pre-processing phase, which includes data cleaning,removing link details, handling missing values, and transforming the data into a suitable format for further processing. All three dataset are used as one.Data has attributes namely text-reviews and ratings.After snorkel sentiment labeling the data is sampled with equal number of all available categories in sentiment label.

3.3 Data Description

The study uses large datasets as BM-amazon-reviews , AR-amazon-product-reviews and amazon-product-review to perform a thorough sentiment analysis.BM-amazon-reviews consist of two folder as train and test dataset.All the three dataset consist of only two attributes as reviews and ratings.Collected reviews improves the computational and analytical capacities of the research even further. The framework for sentiment analysis that captures the complex terrain of online customer feedback across various product domains and review types by combining several datasets with differing dimensions and characteristics.For semantic representation and feature embedding, the research utilizes pre-trained GloVe embeddings (glove.6B.100d.txt), enabling sophisticated linguistic mapping and contextual understanding.

- **BM-amazon-reviews**: Contain product review as 15536 for training data and testing data as 15705 after filtering only product review.
- **AR-amazon-product-reviews**:Has large amount of data whose count is 568454 of product review used for training data.
- **amazon-product-review**:Includes customer reviews for product as 300000

3.4 HARDWARE REQUIREMENTS

3.4.1 Processor (CPU):

- **Model**: Intel Core i7 (10th Gen) or Ryzen 7 3700X.
- **Cores/Threads**:Minimum 8 cores /16 threads.
- **Clock Speed**:3.2 GHz or higher

3.4.2 Memory(RAM):

- **Capacity**: Minimum 16GB
- **Type**:DDR4

3.4.3 Storage:

- **Capacity**: 500 GB SSD
- **Secondary Storage**:1TB HDD

3.4.4 Graphics Processing Unit(GPU):

- **Model**: NVIDIA Geforce GTX 1650 with 4 GB VRAM or equivalent

3.5 SOFTWARE REQUIREMENTS

3.5.1 Programming Language

Python is easy to implement with many libraries like NLTK, Text Blob, and VADER provide pre-trained models, simple APIs, and specialized functionality for analyzing sentiments in textual data.

3.5.2 Tools and libraries

- **Tensorflow**: Build and train deep learning models like BiLSTM
- **Transformers**:Transformer is base for bert
- **Snorkel framework**:Used for weak supervision and auto-labeling of datasets.
- **Numpy**: Does numerical computations and tensor manipulations.
- **Transformers**:Transformer is base for bert
- **Pandas**: Pre-processing and handling dataset.
- **scikit-learn**:For evaluation metrics and data splitting.
- **NLTK**: For text pre-processing and tokenization.
- **Matplotlib**:For visualizing results and performance metrics.
- **Seaborn**:For advanced visualization of evaluation results.

3.6 Feature Extraction

Feature extraction is more important especially for emotional sentiment analysis in e-commerce perspective Major aspect factor chosen namely price, size, service, quality and usability that describe emotion about the product in effective and efficient way. Feature extraction is done by means of auto-labeling algorithm. Aspect Dictionary Construction is done by developing a comprehensive external resource which contains aspect-specific keywords under the list of five aspect as PRICE, SIZE, SERVICE, QUALITY and USABILITY.Create a structured taxonomy of aspect categories by systematic mapping of review text to specific product dimensions. Thus extract aspect keyword to be structured as aspect terms and its respective aspect category for each and every review.

Description of Aspect-Category

- **Price**: Cost-related terms and economic perceptions
- **Size**: Magnitude or dimensions of a thing.
- **Service**: Customer interaction and support-related descriptors
- **Usability**: Functionality and ease of use
- **Quality**: Performance and durability descriptors

3.7 Snorkel labeling

Snorkel is a open source framework which is used for generate label for weak supervision by using labeling function as positive , negative and neutral that define as sentiment classification. Snorkel is a programmatic labeling that enables automatic generation of large scale data that reduce need for manual annotation. Leverages multiple weak supervision sources simultaneously. Snorkel handle labeling complexity by capturing nuanced sentiment beyond binary classification .Snorkel integrates diverse information sources like ratings, review text, and contextual cues.Snorkel generates high-quality training labels at scale by reducing annotation overhead and associated costs.Create a robust, adaptable sentiment labeling approach that transcends traditional manual annotation limitations.Works as defining labeling functions later applied to train labeling model. After training those model is used to predict label for sentiment classification.

3.8 HABSC ALGORITHM

Algorithm creates a sentiment analysis system that examines both overall feelings and specific aspects within reviews.In order to enhance multi model fusion is followed as below

- **VADER**:Valance Aware Dictionary and Sentiment Reasoner which is able to extract emotional expressions that is review about aspects and style of language. Vader is a rule based sentiment analyzer. Based on lexical feature each word has sentiment intensity score between (-4,+4).In sentiment analysis VADER work more effectively as it has ability to handles punctuation , word-shape emphasis, emoji's and contractive conjunctions.Vader score is obtained by normalizing sentiment intensity score between (-1,+1).
- **BERT**: Bidirectional Encoder Representation from

Transformers is used for understanding complex context and modern language patterns. BERT is inferred as pipeline as flow of tokenism , model output , probabilities and at last get prediction.Adds classification layer by this trains on sentiment labeled data to predict the label for concerned review

- **SENTIWORDNET**:NLTK provide parsed version of same corpus. Label each word with its POS (parts of speech).Word dictionary builds on top of the original Princeton WordNet dictionary that adds sentiment score to each word. The scores all add up to 1 where 1 means positive and 0 means negative. Combining all scores as of POS overall average sentiment score is obtained . Hence it follows traditional vocabulary-based sentiment analysis

The algorithm is designed to reduce reliance on a single model by leveraging complementary strengths of VADER, BERT, and SentiWordNet to ensure accurate and context-aware sentiment scoring.As predefined aspect terms is incorporated, it provides targeted insights for specific areas (e.g., price, quality, usability) within a review, enabling businesses to make informed decisions. **FORMULA:**

$$CS = (SWNscore * 0.4) + (Vaderscore * 0.35) + (Bertscore * 0.25) \quad (3.1)$$

3.8.1 Reason for termed as algorithm

Though it uses model like BERT lets see the reason for considering it as algorithm.

- Combine score is obtained as of rule based approach.
- Without training model is used.
- For whole process uses static weights and static thresholds.

3.9 HYBRID BiLSTM MODEL

Traditional unidirectional LSTM networks that process input sequences in a single linear direction.Bi-Directional Long Short-Term Memory (BiLSTM) neural networks architecture process input sequences in both forward and backward directions, enabling a more comprehensive understanding of contextual dependencies.Forward LSTM processes sequence from start to end and backward LSTM processes sequence from end to start.Concatenates outputs from both directions which generates rich, contextually aware representations.Captures sentiment nuances across sequence by that handles complex linguistic structures.Since Bilstm has to perform aspect based sentiment classification hybrid BiLSTM model is trained.Here aspect term and rating are passed as weak signals.Review input is embedded and passed through BiLSTM. The output of BiLSTM and weak signals are

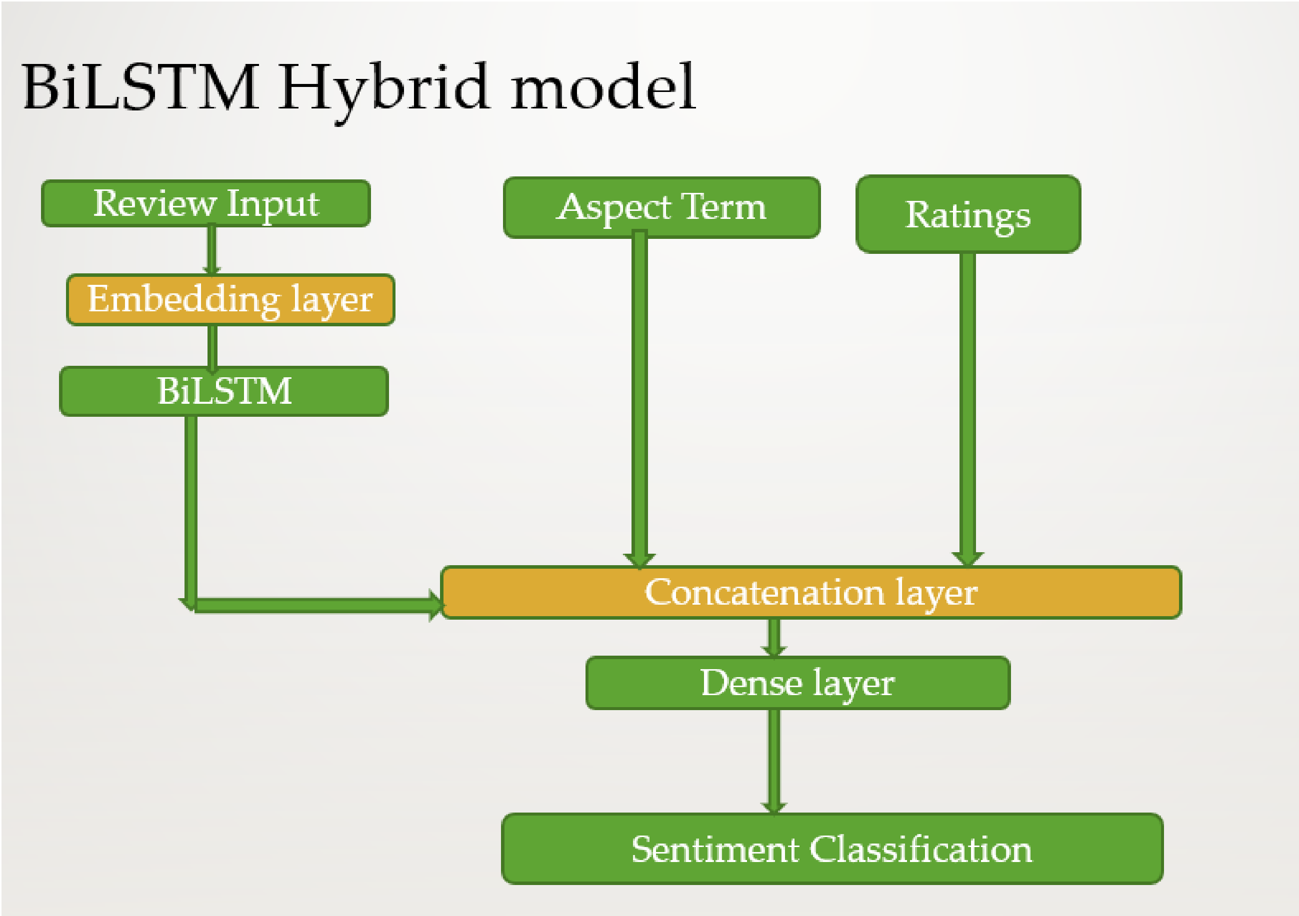

Figure 3.1: HYBRID BI-LSTM MODEL

concatenated and passed through dense layer which fully connected layer, thus sentiment classification is done which will be the final outcome of BiLSTM hybrid model.

3.10 ABSC DeBERT HYBRID MODEL

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) is a revolutionary development in transformer-based language models, building upon and enhancing the foundational BERT architecture.By introducing a complex disentangled attention mechanism that processes position and content information independently, this novel model allows for a more nuanced comprehension of contextual relationships within text sequences. A disentangled attention matrix that takes into account content-to-content, content-to-position, and position-to-content relationships, as well as an improved mask decoder that includes absolute positions in the decoding layer, are two significant innovations that DeBERT adds to the conventional self-attention mechanism. As for aspect based sentiment classification uses hybrid model where inputs are reviews, aspect term ,aspect category and ratings. After being tokenism, reviews and aspect terms are combined to be taught using BERT embedding and an attention mask for semantic understanding.as weak signal aspect category is encoded by label encoding and ratings are embedded before passing.The outcome of BERT, label encoding and embedded rating are concatenated at concatenation layer.Drop out layer is to avoid over fitting then pass on to dense layer and hence finally outcome is obtained by means of filters sentiment classification is done.

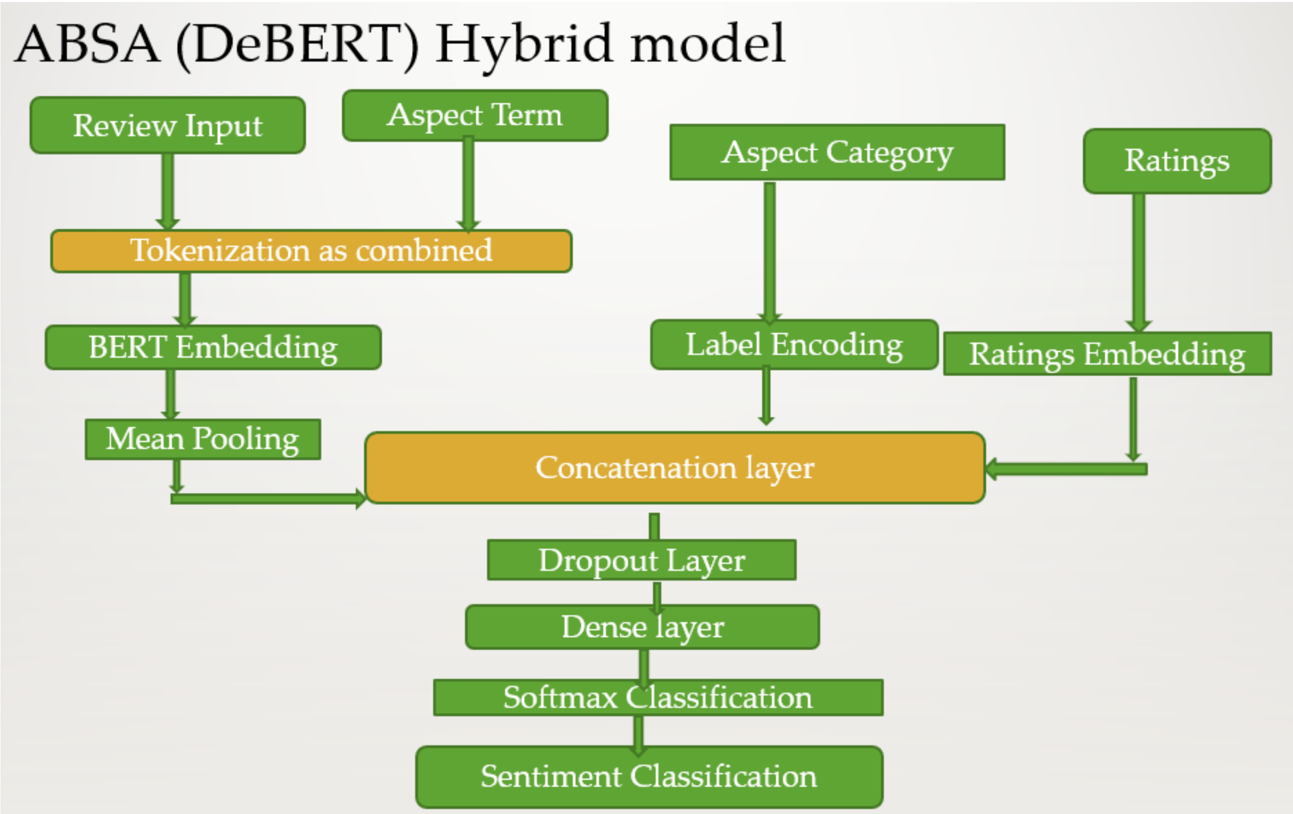

Figure 3.2: DeBERT HYBRID MODEL

CHAPTER 4

IMPLEMENTATION

This chapter describes the detailed implementation of the system for aspect based sentiment analysis, focusing on the tools, data preparation, feature enhancement, and model training.

4.1 INTRODUCTION

Emotional sentiment classification for product review aims to extract sentiment specific to particular aspects within a text, providing granular insights that differs from traditional sentiment analysis.The Hybrid Aspect-Based Sentiment Classification (HABSC) algorithm highlights the strengths of deep learning models and domain-specific sentiment features to address challenges in ABSC. HABSC aims to improve their accuracy of sentiment label prediction. Hybrid BiLSTM used to extract sequential relationships between tokens, essential for understanding nuanced sentiments tied to specific aspects.Addresses long-range dependencies, enabling the model to detect sentiments across complex sentences.Hybrid DeBERT model extract contextual relationships between aspect terms and sentiments with high precision. and handles domain-specific nuances and polysemy effectively.Hybrid BiLSTM and Hybrid DeBERT achieves high-accuracy, context-aware and scalable sentiment analysis.

4.2 Data Pre-processing

- Filter data: As emotional sentiment classification it to be performed only for e-commerce review filter only product review .Remove reviews like music,book,films, story, game etc.
- Concatenate data: In order to increase the number of data concatenate three sets of data using pandas as mentioned in data description.
- Handling Missing Values: Missing values or not available are replaced with none to avoid computational errors.
- Remove noise: Remove links or unwanted data.

4.3 ALGORITHM 1

Auto-Labeling Algorithm

4.3.1 Input:

- **Reviews**: Customer reviews about product as text data.
- **Aspect keyword dictionary**:A dictionary where the keys are aspect names (e.g., price, quality, usability), and the values are lists of predefined keywords related to each aspect.

4.3.2 Procedure:

- **Pre-process**: Convert the review text to lowercase.
- **Check Keyword Matching**: Iterate through the list of keywords associated with each aspect.
- **Store as attribute**: If matches are found, store the matched keywords for that aspect.Populate each column with the matched keywords for the respective aspect from the corresponding review

4.3.3 Output:

Data Frame include additional columns for each aspect (e.g., price, quality, usability, size and service) containing the matched keywords from the reviews.

Algorithm 4.1 Extract Aspect Keywords from Review Text

Require:
train_data: A DataFrame containing a column `Review_Text` (list of review texts).
aspect_keywords: A dictionary where keys are aspect names and values are lists of keywords.

Ensure: Updated *train_data* with aspect-specific columns containing matched keywords.

```
procedure EXTRACTKEYWORDS(review, keywords)
    Initialize found_keywords ← []
    for all keyword in keywords do
        if keyword exists as a whole word in review (case-insensitive) then
            Append keyword to found_keywords
        end if
    end for
    if found_keywords is not empty then
        return Concatenation of found_keywords as a comma-separated string
    else
        return None
    end if
end procedure
for all aspect, keywords in aspect_keywords do
    for all review in train_data[Review_Text] do
        train_data[aspect] ← EXTRACTKEYWORDS(review, keywords)
    end for
end for
```

Aspect Category: Populate an attribute namely aspect category which has name of aspect for concerned review that dealt with.

4.4 ALGORITHM 2

Snorkel Sentiment Labeling Algorithm

Input:

- Train data: Data is split whose attributes as reviews and ratings.
- Test data: Data whose attributes to be considered are reviews and ratings.
- Sentiment Constants: ABSTAIN (-1): No label assigned, POSITIVE (1): Positive sentiment, NEGATIVE (0): Negative sentiment, MIXED (2): Mixed sentiment.
- Sentiment Analyzer:VADER's SentimentIntensityAnalyzer used for analyse to define sentiment score

Procedure

- Define Labeling Functions : If sentiment criteria in reviews and ratings matches assign POSITIVE ,NEGATIVE or MIXED else ABSTAIN.
- Apply Labeling Functions: Use PandasLFApplier to apply the labeling functions to training and testing data, producing label matrices L-train and L-test.
- Train the Label Model: Use Snorkel's Label Model to train on L-train with 3 classes (cardinality=3) and fit model for 1000 epochs.

Output: Trained Snorkel label model that can predict probabilistic labels to classify as positive, negative and neutral.

Algorithm 4.2 Snorkel Sentiment Labeling

Require:

train_data: A DataFrame containing columns `Rating` and `Review_Text`.
sid_obj: A pre-trained sentiment intensity analyzer (e.g., VADER).

Ensure:

Label matrices *L_train* for training data.

```
procedure LABELINGFUNCTIONPOSITIVE(x)
    rating ← x[Rating]
    review ← x[Review_Text]
    sentiment_dict ← sid_obj.POLARITY_SCORES(review)
    if sentiment_dict[compound] ≥ −0.05 and rating = 1 then
        return POSITIVE
    else
        return ABSTAIN
    end if
end procedure
procedure LABELINGFUNCTIONNEGATIVE(x)
    rating ← x[Rating]
    review ← x[Review_Text]
    sentiment_dict ← sid_obj.POLARITY_SCORES(review)
    if sentiment_dict[compound] ≤ −0.05 and rating = 0 then
        return NEGATIVE
    else
        return ABSTAIN
    end if
end procedure
procedure LABELINGFUNCTIONMIXED(x)
    rating ← x[Rating]
    review ← x[Review_Text]
    sentiment_dict ← sid_obj.POLARITY_SCORES(review)
    if sentiment_dict[compound] ≥ −0.05 and rating = 0 then
        return MIXED
    else if sentiment_dict[compound] ≤ −0.05 and rating = 1 then
        return MIXED
    else
        return ABSTAIN
    end if
end procedure
procedure APPLYLABELINGFUNCTIONS
    lfs ← [LABELINGFUNCTIONPOSITIVE, LABELINGFUNCTIONNEGATIVE, LABELIN
    applier ← PANDASLFAPPLIER(lfs)
    L_train ← applier.APPLY(train_data)
    return L_train
end procedure
procedure TRAINLABELMODEL
    Initialize label_model ← LABELMODEL(cardinality = 3, verbose =
True)
    label_model.FIT(L_train, n_epochs = 1000, log_freq = 50)
    return Trained label_model
end procedure
```

Sample Data: Based on sentiment label the data is sample as equal number of data under each category of label as it will enhance the accuracy and improve performance.

4.5 HABSC Algorithm

Multi model is used in HABSC for improving its accuracy. Combined score id obtained by combining al model score in a rule based yet label decision is done in ensemble based. **Input:**

- **Train data:** The attributes given as input are Reviews and ratings and external resources which consist of aspect keywords.

Procedure:

- **Initialize Sentiment Analyzers:**Load the VADER sentiment analyzer, BERT transformer-based sentiment analyzer and define constants for sentiment labels: POSITIVE, NEGATIVE, and NEUTRAL
- **Define Functions:** Define wordnet-pos(tag) which convert POS tags to WordNet format (adjective, verb, noun, or adverb), Get sentiwordnet-score(word, pos-tag) that calculate the SentiWordNet sentiment score for a given word and its

POS tag.Then retrieve all synsets for the word with the specified POS tag along with their positive and negative score and finally return the difference among them.

- **SentiWordNet Score Method:** The system processes review text by tokenizing and POS tagging, then matches predefined aspect keywords to identify relevant terms. For each matched aspect term, it calculates a SentiWordNet score based on the word's POS tag, collecting these in aspect-scores. Finally, it computes the average SentiWordNet score across all matches for each aspect to determine the overall aspect-specific sentiment.

- **Sentiment Score for Reviews:** Calculate the compound VADER score for the review and predict the sentiment using BERT; assign +1 for positive and -1 for negative.

- **Calculate Combined score:**

$$\mathrm{CS} = (\mathrm{avg\_SWN\_score} \times 0.4) + (\mathrm{vader\_score} \times 0.35) + (\mathrm{bert\_score} \times 0.25)$$

- **Ensemble-based Decision Making:**

$$\text{If CS} > 0.25 \text{ or } (\text{VADER\_score} > 0 \text{ and BERT\_score} > 0)$$

$$\text{return POSITIVE.}$$

$$\text{If CS} < -0.25 \text{ or } (\text{VADER\_score} < 0 \text{ and BERT\_score} > 0)$$

return NEGATIVE.

$otherwise, returnNEUTRAL.$

OUTPUT: Aspect or emotional based sentiment label as of positive, negative and neutral. CaptionHABSC Algorithm - Multi moel sentiment classification

Algorithm 4.3 Sentiment Labeling - Initialization and Helper Functions (Part 1)

```
procedure INITIALIZE RESOURCES
    Import libraries: pandas, numpy, nltk, transformers.
    Download NLTK resources: vader_lexicon, sentiwordnet, punkt,
    averaged_perceptron_tagger, and wordnet.
    Initialize VADER: SentimentIntensityAnalyzer.
    Initialize BERT pipeline: pipeline("sentiment-analysis").
    Define sentiment labels: positive, negative, neutral.
end procedure
procedure GETWORDNETPOS(tag)
    if tag starts with 'J' then
        return wn.ADJ.
    else if tag starts with 'V' then
        return wn.VERB.
    else if tag starts with 'N' then
        return wn.NOUN.
    else if tag starts with 'R' then
        return wn.ADV.
    else
        return None.
    end if
end procedure
procedure GETSENTIWORDNETSCORE(word, pos_tag)
    Retrieve synsets from swn.senti_synsets(word, pos_tag).
    if No synsets exist then
        return 0.
    end if
    Compute positive score: Mean of syn.pos_score() over synsets.
    Compute negative score: Mean of syn.neg_score() over synsets.
    return pos_score − neg_score.
end procedure
```

Algorithm 4.4 Sentiment Labeling - Combined Scoring and Labeling (Part 2)

```
procedure CombinedAspectReviewSentiment(x)
    Extract Review_Text and Rating.
    if Review_Text is empty then
        return neutral.
    end if
    Truncate review text to 512 tokens using tokenizer.
    Initialize empty list for aspect_scores.
    for all Aspect-Keyword pair in aspect_keywords do
        if Aspect column in x is NaN then
            Skip to next pair.
        end if
        for all Keywords in the Aspect do
            if Keyword matches in Review_Text then
                Tokenize and POS-tag Review_Text.
                Compute SentiWordNet score using matched words and append to aspect_scores.
            end if
        end for
    end for
    Compute avg_sentiwordnet_score as mean of aspect_scores.
    Compute vader_score using VADER.
    Compute bert_score: 1 for POSITIVE, -1 otherwise.
    Compute combined_score:
```

$$\text{combined\_score} = 0.4 \cdot \text{avg\_sentiwordnet\_score} + 0.35 \cdot \text{vader\_score} + 0.25 \cdot \text{bert\_score}.$$

```
    if combined_score ¿ 0.25 or (vader_score ¿ 0.5 and bert_score = 1) then
        return positive.
    else if combined_score ¡ -0.25 or (vader_score ¡ -0.5 and bert_score = -1) then
        return negative.
    else
        return neutral.
    end if
end procedure
procedure ApplyLabeling(data)
    Apply CombinedAspectReviewSentiment row-wise to data.
    Add results to a new column: sentiment_label.
    return Updated DataFrame.
end procedure
```

4.6 Hybrid BiLSTM Model

A Bidirectional LSTM, or biLSTM, is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. BiLSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm (e.g. knowing what words immediately follow and precede a word in a sentence).

Input:

- **Review Text:** Tokenized text sequences from customer reviews
- **Aspect Terms:** Keywords related to specific product aspects, tokenized and padded to a fixed length.
- **Ratings:** Numerical or categorical ratings of the review (e.g., positive, neutral, or negative).

Procedure:

- **Data Preparation** Pre-process and tokenize the Review-Text column into sequences and Aspect-Terms column for uniform input size.Encode the Rating column into numerical values and one-hot encode for classification.

- **Model Definition**Three input layers for Review-Text, Aspect-Lexicon, and Rating applied to embedding layer and a BiLSTM layer to extract features from the Review-Text.Concatenate the BiLSTM output with the Aspect-lexicon and Rating inputs.Later concatenated features through dense layers to classify sentiment.
- **Train Model** Compile the model with Adam optimizer and categorical cross entropy loss.Train the model using early stopping to prevent over fitting, with a validation split of 20

OUTPUT: Training and validation loss/accuracy for each epoch.The model outputs one of the three classes: positive, neutral, or negative.

Algorithm 4.5 BiLSTM-Based Sentiment Classification

procedure DATA PREPARATION
  Tokenize and pad `Review_Text` to fixed length T_r.
  Tokenize and pad `Aspect_Terms` to fixed length T_a.
  Encode `Ratings` into one-hot vectors Y_r.
  Encode sentiment labels into one-hot vectors Y_s.
end procedure
procedure MODEL DEFINITION
  Define input layer for `Review_Text` with shape $(T_r,)$.
  Define input layer for `Aspect_Lexicon` with shape $(T_a,)$.
  Define input layer for `Ratings` with shape $(1,)$.
  Apply an embedding layer to `Review_Text`.
  Process the embedding output with a BiLSTM layer.
  Concatenate BiLSTM output with `Aspect_Lexicon` and `Ratings`.
  Pass concatenated features through a dense layer with softmax activation for classification.
end procedure
procedure MODEL TRAINING
  Compile model with Adam optimizer and categorical crossentropy loss.
  Set early stopping criterion with patience $p = 3$.
  Train model with validation split of 20%, batch size $b = 32$, and maximum epochs $e = 10$.
end procedure
procedure OUTPUT
  Display model architecture and training metrics.
  Return trained model and sentiment predictions.
end procedure

4.7 ABSC DeBERT Hybrid Model

Input

- **Review Texts** (`reviews`): A column of textual reviews from the dataset.
- **Aspect Terms** (`aspect_terms`): Terms related to specific aspects in the reviews.

- **Aspect Categories** (`aspect_category`): Predefined categories for each review (e.g., *price, service, quality*).
- **Ratings** (`ratings`): Numeric values indicating user ratings.
- **Sentiment Labels** (`debug_labels`): Target labels for sentiment classification.

4.7.1 Procedure

The microsoft/deberta-v3-base model is fine-tuned with additional inputs, including aspect categories and ratings. Mean pooling is applied to BERT outputs, and all features are concatenated before final classification using dense layers.

4.7.2 Output

A trained model for sentiment classification, evaluated on the validation set for accuracy and loss.

Algorithm 4.6 ABSA-DeBERT Hybrid Model (Part 1)

- **Train_data**: Dataset with columns `Review_Text`, `Aspect_Terms`, `Aspect_Category`, `Rating`, `Snorkel_Sentiment_Labels`
- **absa_model_name**: Pretrained model (`microsoft/deberta-v3-base`)
- **max_review_length**: Maximum token length for input reviews

Trained model for sentiment classification
Step 1: Load Tokenizer and ABSA-BERT Model
Load the tokenizer and model using `AutoTokenizer` and `TFAutoModel` from the `transformers` library.
Step 2: Prepare Dataset Columns
Convert `Review_Text`, `Aspect_Terms`, and `Snorkel_Sentiment_Labels` to strings.
Convert `Rating` to floating-point values.
Step 3: Tokenize Inputs
Combine `Review_Text` and `Aspect_Terms` using separator `[SEP]`.
Tokenize the combined texts using the tokenizer with parameters:

- `max_length = max_review_length`
- `padding = "max_length"`
- `truncation = True`
- `return_tensors = "tf"`

Step 4: Encode Aspect Category
Apply `LabelEncoder` to encode `Aspect_Category`.
Create an embedding layer for `Aspect_Category` with an output dimension of 16.
Step 5: Encode Sentiment Labels
Apply `LabelEncoder` to encode `Snorkel_Sentiment_Labels`.
Convert labels to one-hot encoding using `to_categorical`.
Step 6: Process Ratings Input
Create a dense layer with ReLU activation for `Rating` input.
Step 7: Define ABSA-BERT Embedding Function
Use the BERT model to generate `last_hidden_state` embeddings.
Apply mean pooling along the sequence dimension.

Algorithm 4.7 ABSA-DeBERT Hybrid Model (Part 2)

Step 8: Concatenate Features
Concatenate:

- ABSA-BERT embeddings
- Flattened aspect category embeddings
- Ratings dense layer output

Apply dropout to concatenated features.
Step 9: Add Classification Layers
Add a dense layer with ReLU activation.
Add an output layer with 3 units and softmax activation.
Step 10: Compile Model
Compile the model with:

- Optimizer: Adam (learning rate 2×10^{-5})
- Loss: Categorical Crossentropy
- Metrics: Accuracy

Step 11: Prepare Training Data
Define input list:

- `input_ids` from tokenizer
- `attention_mask` from tokenizer
- Encoded aspect categories
- Ratings reshaped as $(\text{batch size}, 1)$

Use one-hot encoded labels for training.
Step 12: Train Model
Train the model for 5 epochs with:

- Batch size: 16
- Validation split: 0.1

End of Algorithm

CHAPTER 5

RESULTS AND COMPARATIVE ANALYSIS

This chapter presents the comparative study of sentiment classification of Snorkel and HABSC algorithm. Compare the outcome of two hybrid model designed based on Bilstm and DeBERT.The result shows BiLSTM hybrid model works the best.

5.1 Model Performance

In this project three hybrid model is used for emotional based sentiment classification.Since there is one HABSC algorithm it is evaluated separately. The BiLSTM and ABSC DeBERT hybrid model are compared.

5.1.1 HABSC Algorithm

HABSC Algorithm is a multi model which define sentiment classification by means of ensemble based decision making. As it follows ensemble based decision making it almost gives equal weight-age to all types of model scores that in turn increase the performance.Performance metrics are evaluated as HABSC sentiment label as predicted label and snorkel sentiment label is being treated as target label that is true value.

Metrics	HABSC
Accuracy	0.85
Precision	0.86
Recall	0.85
F1 Score	0.85

Table 5.1: Comparative Study of Emotional based Sentiment Classification for HABSC Algorithm

Description of evaluation metrics Describe how the performance metrics is being calculated and what it determine with respective meaning.Lets see the explanation about the meaning of metrics result.

- **Accuracy:** Accuracy is the proportion of correct predictions (both positive and negative) made by the model out of all predictions.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}$$

  n this case, an accuracy of 0.85 means that 85 percent of the predictions made by the HABSC model matched the true sentiment labels provided by Snorkel. Accuracy is a general measure of how well the model is performing across all classes (positive, neutral, negative)

- **Precision:** Precision is the proportion of correctly predicted positive sentiment labels out of all predicted positive labels.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

With a precision of 0.86, this means that when the HABSC model predicted a positive sentiment, 86 percent of those predictions were actually positive. Precision focuses on the correctness of the positive predictions and is crucial in situations where false positives are costly (e.g., predicting positive sentiment when it's actually neutral or negative).

- **Recall:** Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive labels that were correctly identified by the model.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

A recall of 0.85 indicates that the HABSC model successfully identified 85 percent of the true positive sentiment labels. Recall is important when the cost of missing a true positive (false negative) is high, such as failing to identify positive reviews.

- **F1 Score:** The F1 Score is the harmonic mean of Precision and Recall. It provides a balance between the two, especially useful when the class distribution is imbalanced.

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

The F1 score of 0.85 suggests a good balance between Precision and Recall. It indicates that the HABSC model is not only precise in its positive sentiment predictions but also good at catching most of the true positive labels. The F1 score is especially useful when we need to balance both false positives and false negatives.

Together, these metrics—each with its own advantages—help to clarify the trade-offs between identifying positive sentiment and making sure forecasts are accurate, which is crucial for sentiment analysis activities.As all metrics has almost same percentage shows the algorithm works well with respect to all positive and negative combinations.

5.2 Comparative analysis of Hybrid model

Comparing these two models allows us to evaluate whether the computational complexity and advanced pre training of DeBERTa translate into significant performance improvements over BiLSTM in the context of sentiment analysis. A comparison reveals the hybrid design's effectiveness for each model and its ability to leverage external features alongside text data. **Hybrid BiLSTM Model** The BiLSTM component serves as an efficient memory-conscious solution that processes text in dual directions, making it particularly good at understanding how words influence each other within sentences. Its lightweight nature means it can run smoothly on standard hardware, and it pairs especially well with domain-specific training data and vocabulary. Think of it as a specialized tool that's perfect for focused, resource-efficient analysis.BiLSTM can explicitly encode sequential features and is easier to combine with handcrafted features (e.g., aspect scores). BiLSTM-based models may work well for simpler tasks or smaller datasets where the additional power of transformers is not necessary. BiLSTM might perform well for domain-specific datasets if trained with domain-relevant embeddings and features. **Hybrid DeBERT Model** DeBERTa uses pre-trained contextual embeddings, which may interact differently with external inputs like aspect terms. In contrast, the DeBERTa architecture represents a more sophisticated approach to language understanding. By separating how words relate to each other from where they appear in a sentence, it achieves a deeper grasp of meaning. Its pre-exposure to vast amounts of text data means it starts with a rich understanding of language, making it especially powerful for complex sentiment patterns and subtle meaning variations. However, this power comes with higher computational demands.DeBERTa, being pre-trained on diverse datasets, might generalize better across multiple domains and languages. DeBERTa-based models are expected to excel in tasks with complex semantics and larger datasets due to their pre-training and ability to

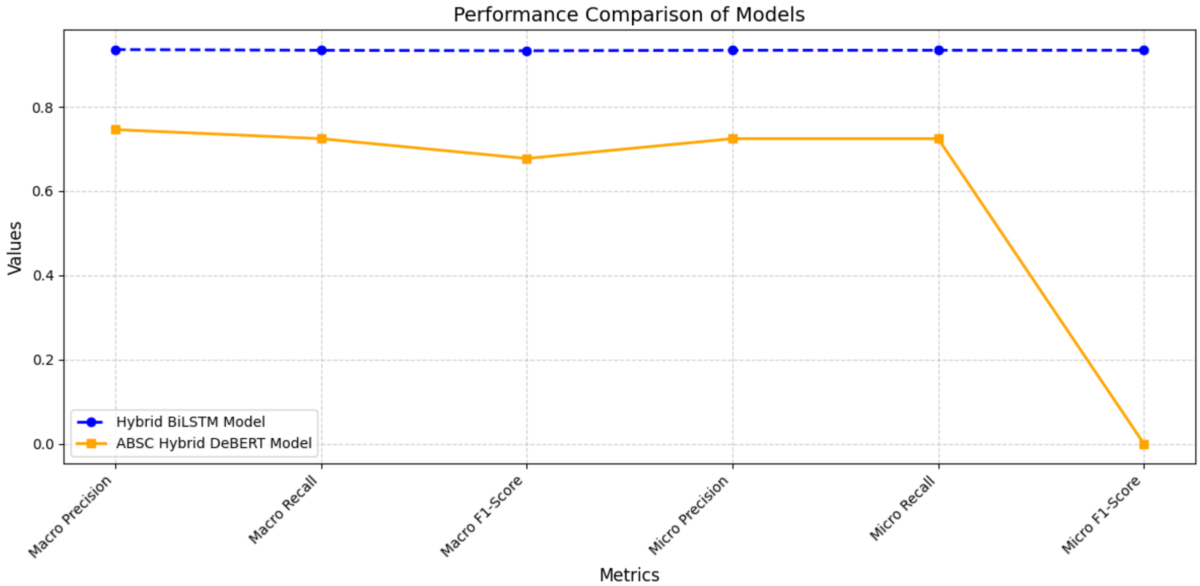

Figure 5.1: Performance study of two hybrid model

model context at a deeper level.

Table 5.2: Comparison of Hybrid BiLSTM and ABSC DeBERT Models

Metrics	Hybrid BiLSTM Model	Hybrid ABSC DeBERT Model
Macro Precision	0.9358	0.7459
Macro Recall	0.9341	0.7243
Macro F1-Score	0.9330	0.6773
Micro Precision	0.9342	0.7243
Micro Recall	0.9342	0.7243
Micro F1-Score	0.9342	0.7243
Hamming Loss	0.0659	0.2757

Description of Performance Metrics In this section describes the performance metrics and explanation what it means.Lets see the gest of the metrics.

- **Macro Precision:** Evaluates the precision (true positives / all predicted positives) for each class independently and averages them without considering the class imbalance.

$$\text{Macro Precision} = \frac{1}{C}\sum_{i=1}^{C}\text{Precision}_i$$

- **Macro Recall:** Evaluates the recall (true positives / all actual positives) for each class independently and averages them by that measures the model's ability to identify all relevant instances for each class.

$$\text{Macro Recall} = \frac{1}{C}\sum_{i=1}^{C}\text{Recall}_i$$

- **Macro F1-Score:** Harmonic mean of macro precision and macro recall.Provides a balanced metric for model performance across all classes, regardless of class distribution.

$$\text{Macro F1-Score} = \frac{1}{C}\sum_{i=1}^{C}\text{F1-Score}_i$$

- **Micro Precision, Recall, and F1-Score:** These metrics consider the overall true positives, false positives, and false negatives across all classes, treating all instances equally. Useful when class imbalance exists, as they focus on global performance.

$$\text{Micro Metric} = \frac{\text{TP}_{\text{total}}}{\text{TP}_{\text{total}} + \text{FP}_{\text{total}} + \text{FN}_{\text{total}}}$$

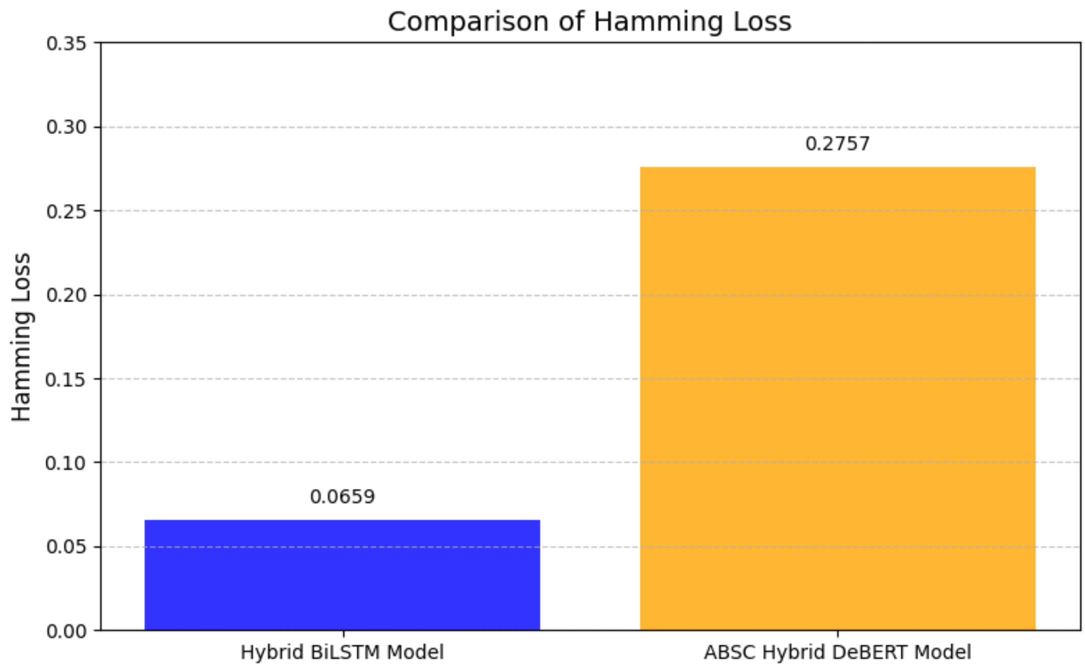

Figure 5.2: Comparison of hamming loss

- **Hamming Loss:** The fraction of incorrectly predicted labels. Lower values indicate better model performance, as it signifies fewer errors in predictions.

$$\text{Hamming Loss} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{Number of incorrect labels for } i}{\text{Total number of labels for } i}$$

Conclusion:

- **Hybrid BiLSTM Model:** Exhibits higher Macro and Micro Precision, Recall, and F1-Score, indicating superior handling of both class-level and instance-level performance. The low Hamming Loss further confirms its effectiveness in minimizing prediction errors.

- **Hybrid ABSC DeBERT Model:** Lower metrics suggest challenges in leveraging the pre-trained model for aspect-based sentiment analysis. The higher Hamming Loss reflects more frequent label prediction errors.

The Hybrid BiLSTM model demonstrates stronger performance due to its ability to effectively model sequential relationships and integrate aspect-specific features. While the ABSC DeBERT model offers potential advantages in generalization, its performance lags in this specific task.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Emotion based sentiment analysis in e-commerce review will provide better understanding of product advantage and disadvantage.Explored three advanced models for emotion-based sentiment analysis (ABSA): HABSC, Hybrid BiLSTM, and Hybrid DeBERT, leveraging state-of-the-art techniques to improve sentiment classification and emotional understanding in reviews.

6.1 Conclusion

The surveyed literature highlights various approaches to enhancing performance particularly through BiLSTM and BERT for aspect based sentiment classification. The HABSC Algorithm comprises the BERT score described as context-aware sentiment scoring, the SentiWordNet Score is based on word definitions and usage, the VADER score is a lexicon- and rule-based component by combining and ensemble-based decision making leading to better performance. As for the hybrid BILSTM and ABSC Hybrid DeBERT model, the BiLSTM model performs better. Captured long-term dependencies and semantic nuances in textual data with a deeper contextual understanding of aspects (emotions).This makes the model to better understand emotions in customer review from the e-commerce perspective. The hybrid BiLSTM model emerged as the best performer in terms of accuracy, precision and F1-Score, making it the ideal choice for applications that prioritize high predictive accuracy and nuanced semantic understanding.

6.2 Future Works

As the proposed model works better performance next is to generate model such that it able to predict aspect(emotion) based sentiment classification as well aspect category to provide more insights.Other future work can be as follow :

- **Advanced Weak Supervision Techniques** Refine the weak supervision process by incorporating domain-specific rules or using generative models to improve labeling quality. Explore self-supervised approaches to leverage unlabeled data for pre-training aspect-aware sentiment models.
- **Model Enhancements** ombine transformer-based architectures (like DeBERTa) with recurrent neural networks (like BiLSTM) to exploit both contextual and sequential dependencies. xtend models to simultaneously predict overall sentiment and aspect-specific categories for better efficiency and performance.develop interpretability methods such as AI that can explain predictions at the aspect level, increasing trust in model output.

- **Reinforcement Learning for Aspect Weights** Use reinforcement learning (RL) to dynamically assign weights to aspect-specific scores, improving overall sentiment classification accuracy.Explore reward functions that incorporate both predictive accuracy and explainability.
- **Graph Neural Networks (GNNs)** Leverage GNNs to model relationships between aspects, terms, and sentiments, particularly in reviews with complex interdependencies. Use GNN-based attention mechanisms to improve aspect-level sentiment extraction.

APPENDIX A

TOPIC 1

A.1 SECTION 1

A.2 SECTION 2

APPENDIX B

TOPIC 2

B.1 SECTION 1

B.2 SECTION 2

REFERENCES

[1] Kalpa Subbaiaha . Aspect category learning and sentimental analysis using weakly supervised learning. *IEEE Transactions on Information Theory*, 49:2017–2309, 2022.

[2] Zhou. A brief introduction to weakly supervised learning. *Journal of Economic History*, 5:468–493, 2022.

[3] WACHIRAWUT THAMVISET NIKORN KANNIKAKLANG and SARTRA WONGTHANAVASU. Bilstcan: A novel srs-based bidirectional long short-term capsule attention network for dynamic user preference and next-item recommendation. *IEEE*, 55:3577–3593, 2023.

[4] Unknown. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 59, pages 504–510, Online, August 2021. Association for Computational Linguistics.

[5] Anping Zhao and Yu Yu. Knowledge-enabled bert for aspect-based sentiment analysis. *Unknown*.

[6] Baris Ozyurt and M. Ali Akcayol. A new topic modeling based approach for aspect extraction in aspect-based sentiment analysis: Ss-lda. *Unknown*.

[7] MengChu Zhou Xiaoyu Sean Lu Abdullah Abusorrah Haoyue Liu, Ishani Chatterjee. Aspect-based sentiment analysis: A survey of deep learning methods. *IEEE Transactions on Computational Social Systems*, 7:1358–1375, 2020.

[8] Yang Deng Lidong Bing Wai Lam Wenxuan Zhang, Xin Li. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 35:11019–11038, 2023.

[9] Lianwei Wu Ling Sun Ambreen Nazir, Yuan Rao. Issues and challenges of aspect-based sentiment analysis: A comprehensive survey. *IEEE Transactions on Affective Computing*, 13:845–863, 2022.

[10] Lin K. Sentiment analysis algorithm based on bert and convolutional neural network. 2021.

[11] Richard E Fikes and Nils J Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI'71, pages 608–620, 1971.