ASPECT BASED SENTIMENT ANALYSIS

A Project report submitted in partial fulfillment of the requirements for the award of the Degree of ***Bachelors of Technology***

In

Computer Science and Engineering (CSE)

By

Pingili Shivani Reddy (21011A0536)

Power Vijaya (21011A0538)

Talari Varun (22015A0515)

Under the guidance of

Dr. O. B. V. Ramanaiah

Senior Professor

Department of Computer Science and Engineering,

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

University College of Engineering, Science & Technology Hyderabad

Kukatpally, Hyderabad - 500 085.

Department of Computer Science and Engineering,

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

University College of Engineering, Science & Technology Hyderabad

Kukatpally, Hyderabad - 500 085.

DECLARATION BY THE CANDIDATES

We, **Pingili Shivani Reddy(21011A0536), Power Vijaya(21011A0538), Talari Varun(22015A0515)** hereby declare that the project report entitled “**ASPECT BASED SENTIMENT ANALYSIS”,** developed by our group under the guidance of **Dr. O.B.V. Ramanaiah,** is submitted in partial fulfillment of the requirements for the award of the degree of *Bachelors of Technology* in *Computer Science and Engineering*. This is a record of bona fide work carried out by our group and the results embodied in this project have not been reproduced/copied from any source.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

Pingili Shivani Reddy (21011A0536)

Power Vijaya (21011A0538)

Talari Varun (22015A0515)

Department of Computer Science and Engineering,
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
University College of Engineering, Science & Technology Hyderabad
Kukatpally, Hyderabad - 500 085.

CERTIFICATE BY THE SUPERVISOR

This is to certify that the project report entitled "**Aspect Based Sentiment Analysis",** being submitted by **Pingili Shivani Reddy(21011A0536), Power Vijaya(21011A0538), Talari Varun(22015A0515)**, in partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Computer Science and Engineering,* is a record of bona fide work carried out by them. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

Dr. O. B. V. Ramanaiah,
Senior Professor

Date:

Department of Computer Science and Engineering,

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

University College of Engineering, Science & Technology Hyderabad

Kukatpally, Hyderabad - 500 085.

CERTIFICATE BY THE HEAD OF DEPARTMENT

This is to certify that the project report entitled "**Aspect Based Sentiment Analysis",** being submitted by **Pingili Shivani Reddy(21011A0536), Power Vijaya(21011A0538), Talari Varun(22015A0515),** in partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Computer Science and Engineering* is a record of bona fide work carried out by them.

Dr. K.P. Supreethi

Professor & Head of CSE

Date:

ACKNOWLEDGEMET

We wish to extend our sincere gratitude to our supervisor, **Dr. O.B.V. Ramanaiah**, Senior Professor of the Department**,** for being a driving force all through the way. This project would not be so smooth and so interesting without his encouragement.

We are extremely grateful to **Dr. K. P. Supreethi**, Professor & Head of CSE for her immense support and cooperation that contributed a lot in the completion of the task. We show gratitude to our beloved **Dr. G.V. Narasimha Reddy**, Professor & Principal for providing necessary infrastructure and resources for the accomplishment of our project report at JNTUH University College of Engineering, Science and Technology, Hyderabad. We also thank all the staff members of Computer Science & Engineering Department, JNTUH University College of Engineering, Hyderabad for their valuable support and generous advices.

Finally thanks to our parents, all our family members and friends for their continuous support and enthusiastic help.

Pingili Shivani Reddy (21011A0536)

Power Vijaya (21011A0538)

Talari Varun (22015A0515)

ABSTRACT

Aspect-Based Sentiment Analysis (ABSA) is a technique in natural language processing (NLP) that we used to identify sentiment at the granular level of specific aspects or features of a product. Unlike traditional sentiment analysis, which classifies entire texts as positive, negative, or neutral, we focused on identifying sentiments associated with specific aspects mentioned within a review. For instance, in a product review, aspects like "design" or "power consumption" were analysed separately, and we determined the sentiment (positive, negative, or neutral) towards each aspect. This detailed feature analysis helped understand customer opinions in depth, providing valuable insights for businesses.

The process of ABSA involved several key steps: aspect extraction, sentiment detection, and classification. We first extracted the features or attributes discussed in the text, such as "design" or "power consumption." with 93% accuracy. Next, we detected the sentiment polarity expressed towards these aspects, categorizing them into positive, negative, or neutral classes with 81% accuracy. Finally, we classified the sentiment accordingly, enabling businesses to understand specific strengths and weaknesses of their products. Through this in-depth analysis, we aimed to help companies improve specific product features and address customer concerns more effectively, ultimately leading to enhanced customer satisfaction and better product development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter-1

INTRODUCTION

1. INTRODUCTION

1.1 Motivation

Sentiment analysis, the process of identifying and interpreting emotions expressed in text, has emerged as a critical application in natural language processing (NLP). It plays a pivotal role in helping brands analyse customer feedback, such as opinions from survey responses, product reviews, and social media conversations. By understanding customer sentiments, businesses can tailor their products and services to better meet customer needs, ultimately enhancing customer satisfaction and driving growth. However, traditional sentiment analysis often falls short as it provides only a generalized sentiment polarity for an entire text, overlooking the nuanced opinions expressed about specific entities or aspects within the text.

To address this limitation, Aspect-Based Sentiment Analysis (ABSA) has been introduced as a more advanced and granular approach. ABSA focuses on identifying sentiments associated with specific aspects or features mentioned in the text, such as "battery life" in a smartphone review or "service quality" in a restaurant feedback. This fine-grained analysis enables businesses to gain deeper insights into customer opinions, pinpointing exactly what customers appreciate or dislike about their products or services. By leveraging ABSA, companies can make data-driven decisions to improve specific features, address customer concerns, and enhance overall product quality. This project explored and implemented ABSA technique to provide actionable insights for businesses, bridging the gap between customer feedback and product improvement.

1.2 Problem Definition

Traditional sentiment analysis provides an overall sentiment polarity for a text but fails to capture the nuanced opinions expressed about specific aspects or features within it. This limitation makes it difficult for businesses to understand detailed customer feedback about particular attributes of their products or services, such as "battery life" in electronics or "service quality" in hospitality. To address this gap, Aspect-Based Sentiment Analysis (ABSA) has emerged as a solution, enabling the identification and analysis of sentiments associated with specific aspects. This project focuses on implementing ABSA to provide businesses with granular insights into customer opinions, helping them improve specific product features and enhance customer satisfaction.

1.3 Software and Hardware Specification

Software:

Operating system : Windows 11
Libraries and Frameworks : transformers, numpy, pandas, matplotlib, warnings, Seaborn, sklearn, tensorflow, torch, tqdm
Development Tools : Visual Studio Code, Git

Hardware:

System : Desktop or Laptop
Processor : 2 GHz dual-core processor or higher
RAM : 4 GB+
ROM : 128GB+

1.4 Report Organization

Chapter 2 Provides an overview of the Literature Survey.

Chapter 3 Gives the Use Case Diagram.

Chapter 4 Gives the System Design.

Chapter 5 Provides the procedure of implementation.

Chapter 6 Deals with Metrics and results of the proposed model.

Chapter 7 Gives conclusion deducted from the results followed by references.

Chapter-2

LITERATURE

2. LITERATURE

2.1 Overview of Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a subfield of natural language processing (NLP) that focuses on identifying and extracting subjective information from text to determine the sentiment expressed. It involves analysing textual data, such as reviews, social media posts, or survey responses, and classifying the content into categories like positive, negative, or neutral based on the emotional tone. This technique is widely used across industries to gauge public opinion, monitor brand reputation, and understand customer feedback. By automating the process of sentiment detection, businesses can efficiently analyse large volumes of text data and derive actionable insights to improve their products, services, and customer experiences.

The importance of sentiment analysis lies in its ability to provide a quick and scalable way to understand human emotions and opinions at scale. It is particularly useful in areas like market research, customer support, and social media monitoring, where understanding public sentiment is critical for decision-making. For example, companies can use sentiment analysis to track customer satisfaction, identify emerging trends, or detect potential issues before they escalate. Despite its simplicity, sentiment analysis serves as a foundational tool in NLP, enabling organizations to harness the power of textual data for strategic and operational improvements.

2.2 Aspect Based Sentiment Analysis

Aspect-Based Sentiment Analysis (ABSA) is an advanced technique in natural language processing (NLP) that focuses on identifying and analysing sentiments associated with specific aspects or features mentioned in a text [1]. Unlike traditional sentiment analysis, which provides an overall sentiment for the entire text, ABSA breaks down the text to detect sentiments related to individual aspects, such as "battery life" in a product review or "service quality" in a restaurant feedback. This granular approach allows businesses to gain deeper insights into customer opinions, helping them understand which aspects of their products or services are performing well and which need improvement. ABSA is particularly useful in domains like e-commerce, hospitality, and customer feedback analysis, where understanding detailed customer sentiments is crucial for decision-making.

The process of ABSA typically involves two main tasks: **aspect**

extraction and **sentiment classification**. Aspect extraction identifies the specific features or attributes discussed in the text, such as "design" or "performance," while sentiment classification determines the sentiment (positive, negative, or neutral) expressed toward each aspect. By combining these tasks, ABSA provides a comprehensive view of customer feedback, enabling businesses to address specific concerns and enhance customer satisfaction. For example, a smartphone review might praise the "camera quality" but criticize the "battery life," and ABSA helps businesses pinpoint these detailed insights. This fine-grained analysis makes ABSA a powerful tool for improving product development, customer experience, and overall business strategies.

2.3 Challenges in ABSA

Domain Dependency:
Aspect-Based Sentiment Analysis (ABSA) models often struggle with domain-specific language and terminology. A model trained on restaurant reviews may not perform well on electronics reviews, requiring domain adaptation or retraining, which can be time-consuming and resource-intensive.

Data Sparsity:
Aspect-level sentiment analysis requires labelled data for specific aspects, which is often scarce or unevenly distributed. This lack of sufficient annotated data makes it challenging to train robust and accurate ABSA models.

Aspect Extraction Accuracy:
Identifying relevant aspects in a text can be difficult, especially when they are implicitly mentioned or phrased in diverse ways. Inaccurate aspect extraction can lead to incorrect sentiment classification, reducing the overall effectiveness of the system.

Context Understanding:
Sentiment toward an aspect can depend heavily on context. For example, the phrase "not bad" might imply a positive sentiment, but the negation makes it challenging for models to interpret correctly. Capturing such nuances is a significant challenge in ABSA.

Handling Multiple Aspects:
Texts often discuss multiple aspects simultaneously, and sentiments toward these aspects can vary. Disentangling and accurately classifying sentiments for each aspect without confusion remains a complex task.

Real-Time Processing:
For applications like live customer feedback analysis, ABSA systems need to process data in real-time. Achieving high accuracy while maintaining speed and efficiency is a significant technical challenge.

Chapter-3

USE CASE DIAGRAM

3. USE CASE DIAGRAM

3.1 Over view of Use Case Diagram

A use case diagram is a visual representation in UML (Unified Modeling Language) that outlines the functional requirements of a system by depicting interactions between actors (users or external systems) and the system's key functionalities (use cases). It serves as a high-level requirements blueprint, showing who interacts with the system and what goals they can achieve, without delving into technical implementation details. The diagram typically includes actors (stick figures), use cases (ovals), and relationships (lines/arrows) between them as shown in Figure 3.1.

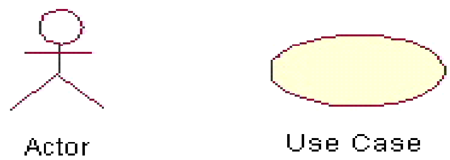

Figure 3.1 Actor and Use case

An actor represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.
Use cases are used in almost every project. These are helpful in exposing requirements and planning the project. During the initial stage of a project most use cases should be defined, but as the project continues more might become visible.

3.2 Diagram Representation

This use case in Figure 3.2 describes the functionality of the Aspect Based Sentiment Analysis.

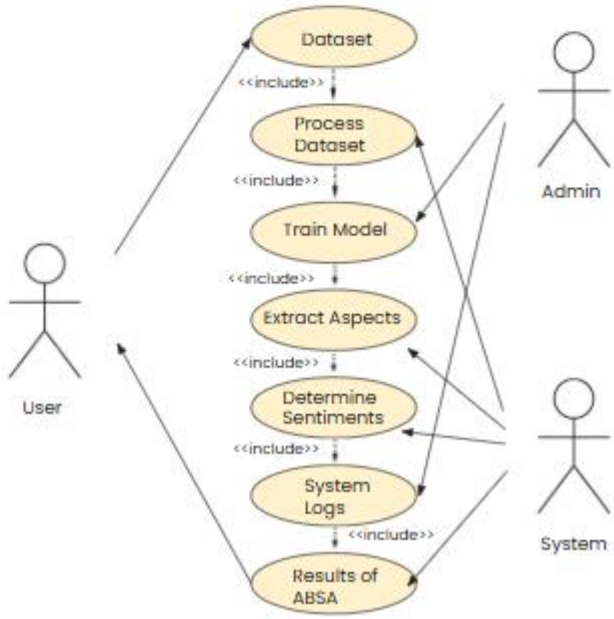

Figure 3.2 Use-Case Diagram

User (End User):
Dataset: Provides raw input (e.g., "Food was great but service slow").
Results of ABSA: Receives structured output (aspect-sentiment pairs + visualizations).

Admin (Technical Team):
Train Model: The admin trains or updates the sentiment analysis model to improve its accuracy.
System Logs: The admin monitors system activity and identifies potential issues.

System (Automated Processes):
Process Dataset: The system receives the user's dataset and initiates the preprocessing.
Extract Aspects: The system identifies the specific features or aspects mentioned in the review (e.g., "battery life," "screen quality," "customer service").
Determine Sentiment: For each extracted aspect, the system determines the sentiment expressed (e.g., positive, negative, neutral).
Results of ABSA: The system stores the extracted aspects and their corresponding sentiment scores and creates visuals like word cloud..

Chapter-4

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 Architecture

The proposed architecture for Aspect-Based Sentiment Analysis (ABSA) leverages a fine-tuned BERT model in a multi-task learning framework to simultaneously perform aspect term extraction and sentiment classification.

4.1.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) [2] is a transformer-based deep learning model designed for natural language processing (NLP). Unlike previous models that processed text sequentially (left-to-right or right-to-left), BERT uses bidirectional attention, meaning it analyzes words in relation to all other words in a sentence simultaneously. This allows it to capture contextual meaning more effectively.

Key Components of BERT:

Transformer Architecture:
Uses self-attention mechanisms to weigh the importance of different words in a sentence.
Consists of multiple encoder layers (12 in bert-base, 24 in bert-large).

Pre-training Tasks:
Masked Language Modeling (MLM): Randomly masks 15% of words and predicts them based on context.
Next Sentence Prediction (NSP): Determines if two sentences logically follow each other.

Tokenization:
Uses WordPiece tokenization to split words into subwords (e.g., "unhappiness" → "un", "happiness").

Adds special tokens:
[CLS] (classification token) for sentence-level tasks.
[SEP] (separator token) for sentence pairs.

Fine-tuning:
Adapts the pre-trained model to specific tasks (e.g., sentiment analysis, question answering) by updating all parameters.

See Figure 4.1 below for a summary of the model architecture for fine-tuning:

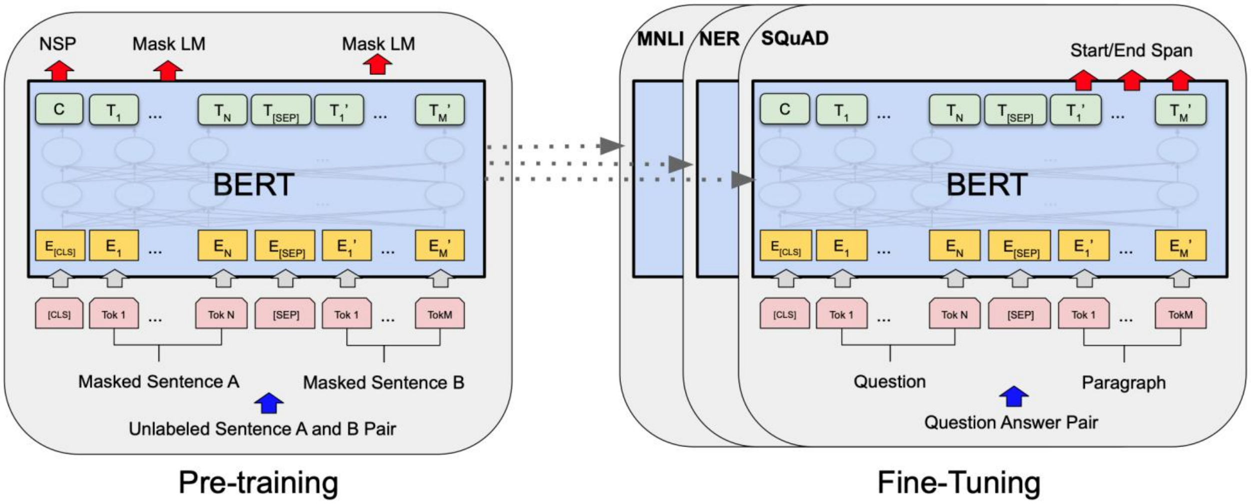

Figure 4.1 BERT Architecture

Reference: https://medium.com/data-science/what-exactly-happens-when-we-fine-tune-bert-f5dc32885d76

4.1.2 BERT on ABSA

For Aspect-Based Sentiment Analysis (ABSA), BERT is fine-tuned in two stages:

Aspect Term Extraction (ATE):
A token classification head is added to predict tags (0-non-aspect, 1- beginning of aspect terms, 2- marks continuation of aspect terms) for each word, identifying aspect terms (e.g., "battery life").

Aspect Sentiment Classification (ASC):
A sequence classification head takes the [CLS] token and aspect-specific embeddings to predict sentiment (positive/negative/neutral) for each extracted aspect.

By fine-tuning BERT on domain-specific data (e.g., product reviews), the model learns to associate aspects with their sentiment polarities, enabling precise, granular sentiment analysis. The bidirectional context helps resolve ambiguities (e.g., "The screen is great, but the battery is poor"), where different aspects have opposing sentiments.
This approach outperforms traditional methods by leveraging BERT's deep contextual understanding while being adaptable to various domains.

4.2 Work Flow Diagram

The ABSA process starts by defining the project's goals and gathering raw text data from sources like reviews or social media. Next, the data is cleaned and preprocessed to remove noise and standardize the format. The system then identifies key aspects (like "battery" or "screen" in a phone review) and analyzes the sentiment (positive, negative, or neutral) associated with each one. Finally, the results are visualized using charts or dashboards to highlight trends and insights, making it easy to understand customer opinions at a granular level. This end-to-end workflow helps businesses pinpoint strengths and weaknesses in their products or services.

1. Start: Define the project goals and scope.
2. Data Gathering: Collect raw text from sources like reviews or social media.
3. Data Preprocessing: Clean and format the text for analysis.
4. Aspect Extraction: Identify key features or topics in the text.
5. Sentiment Analysis: Determine if opinions on each aspect are positive, negative, or neutral.
6. Visualization: Display results in charts or dashboards for easy understanding.
7. End: Finalize insights and conclude the analysis.

See below Figure 4.2 that shows the work flow diagram of aspect based sentiment analysis.

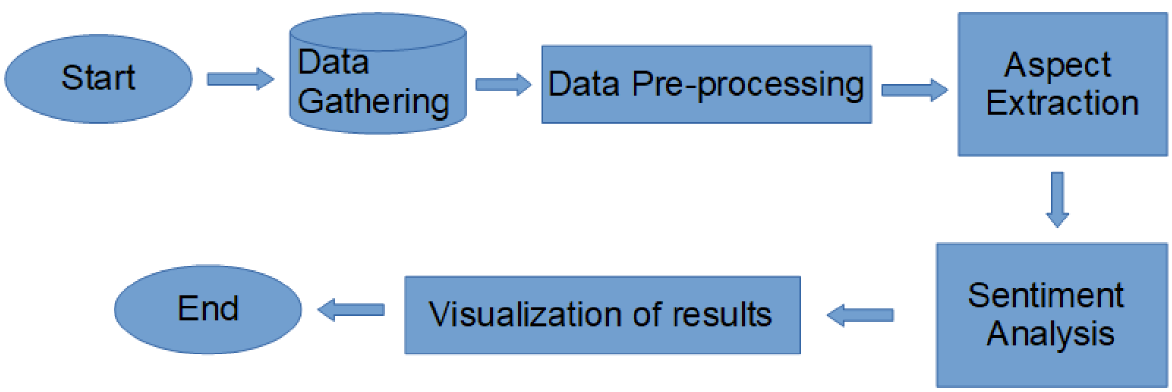

Figure 4.2 Work flow diagram

4.3 Modules

4.3.1 NLTK (Natural Language Toolkit)

The NLTK library played a fundamental role in our text preprocessing pipeline. We utilized its comprehensive suite of tools for tokenization (breaking text into words/sentences), stemming (reducing words to root forms), and lemmatization (proper dictionary reduction of words). The library's part-of-speech tagging capabilities helped identify grammatical structures, while its stopword removal functionality filtered out common but uninformative words. We also employed NLTK's sentiment analysis tools for baseline comparisons and its word frequency analyzers for preliminary data exploration.

4.3.2 NumPy & Pandas

NumPy provided the mathematical backbone for our operations, enabling efficient numerical computations and array manipulations essential for handling word embeddings and model outputs. Pandas was indispensable for data wrangling - its DataFrame structures allowed us to clean, filter, and organize our textual datasets effectively. We used Pandas for merging multiple data sources, handling missing values, and preparing the structured inputs needed for model training. The seamless integration between these libraries ensured smooth data flow through our pipeline.

4.3.3 Transformers (Hugging Face)

The Transformers library gave us access to state-of-the-art BERT models and their variants. We leveraged its pre-trained models (like bert-base-uncased) and tokenizers that properly handled the WordPiece tokenization BERT requires. The library's easy-to-use interfaces allowed us to fine-tune models for our specific ABSA tasks, with built-in support for attention masks and segment embeddings. We particularly benefited from its implementation of transformer architectures and pre-processing tools that maintained consistency with the original BERT paper specifications.

4.3.4 TensorFlow & PyTorch

These deep learning frameworks formed the engine of our model development. TensorFlow's Keras API provided high-level abstractions for building and training our models, while PyTorch offered flexibility for custom architectures. We used their automatic differentiation capabilities for backpropagation during fine-tuning, and their GPU acceleration significantly sped up training. The frameworks' built-in optimizers (like AdamW) and loss functions were crucial for model convergence, and their checkpointing systems enabled us to save and resume training sessions.

4.3.5 TQDM

TQDM's progress bars brought visibility to our lengthy training processes. We integrated it with our data loading and training loops to monitor epoch progress, batch processing, and evaluation metrics in real-time. The library's customizable displays showed estimated completion times, current metrics, and processing speeds, which helped us identify bottlenecks. It proved particularly valuable during hyperparameter tuning when we needed to compare multiple training runs.

4.3.6 Scikit-learn

Scikit-learn provided the evaluation framework for our models. We used its comprehensive suite of metrics - precision, recall, F1-score, and confusion matrices - to rigorously assess model performance on both aspect extraction and sentiment classification tasks. The library's utilities for data splitting (train-test-validation) ensured proper evaluation protocols, and its standardization tools helped preprocess numerical features when needed. We also utilized its implementations of baseline models for comparative analysis.

4.3.7 WordCloud

WordCloud transformed our textual analysis into intuitive visualizations. By generating frequency-based word clouds, we could quickly identify the most prominent aspects in our datasets. We customized the visualizations with color schemes that reflected sentiment polarities (e.g., green for positive aspects, red for negative). These visualizations served both as diagnostic tools during development and as presentation aids when communicating results to stakeholders, making complex textual patterns immediately apparent.

Chapter-5

IMPLEMENTAION

5. IMPLEMENTATION

5.1 Dataset Description

The dataset used for training and evaluating our Aspect-Based Sentiment Analysis (ABSA) model consists of restaurant reviews from the SemEval-2014 ABSA Task [3]. This dataset is structured in a CSV file with three key columns, each designed to facilitate aspect term extraction (ATE) and aspect sentiment classification (ASC). Below is a detailed breakdown of the dataset's structure and annotations:

1. **Sentence**
Description: The raw text of the restaurant review.
Example: "But the staff was so horrible to us."
Purpose: Serves as the input text for aspect extraction and sentiment analysis.
2. **Aspect Term**
Description: The specific feature or entity in the sentence that the sentiment refers to.
Examples: "staff", "food", "atmosphere", "fried rice".
Purpose: Identifies the target of sentiment analysis (e.g., to distinguish between opinions about "food" vs. "service").
3. **Polarity**
Description: The sentiment associated with the aspect term.
Values:
positive (e.g., "amazing", "good"),
negative (e.g., "horrible"),
(Optional) neutral if applicable.
Purpose: Classifies the sentiment toward each aspect term for granular analysis.
Below Table 5.1 shows the sample dataset.

Sentence	Aspect Term	polarity
But the staff was so horrible to us.	staff	negative
Nevertheless the food itself is pretty good.	food	positive
The design and atmosphere is just as good.	design	positive
The design and atmosphere is just as good.	atmosphere	positive
The fried rice is amazing here.	fried rice	positive

Table 5.1 Sample Dataset

5.2 Preprocessing Data

The original dataset (in table format with Sentence, Aspect Term, and Polarity) was transformed into a tokenized, tagged structure (Tokens, Tags, Polarities) to train BERT-based models for Aspect Term Extraction (ATE) and Aspect Sentiment Classification (ASC). First, sentences were tokenized using WordPiece tokenization to split text into subwords. Aspect terms were then labeled using the BIO scheme (Beginning, Inside, Outside), where multi-word aspects like "fried rice" were marked with sequential tags (B=1, I=2). Sentiment polarities were mapped to aspect-bearing tokens (0=negative, 1=neutral, 2=positive, -1=non-aspect tokens), ensuring alignment with the tagged terms. For multi-word aspects, subtokens generated by WordPiece (e.g., "amazing" → ["amaz", "##ing"]) were assigned the same label as their root token to maintain consistency. This structured preprocessing enabled the dataset to train BERT for both aspect extraction (via BIO tags) and sentiment classification (via polarity labels)
The below is the data's structure formatted after preprocessing the dataset:

1. **Column: Tokens**
Description: Contains tokenized sentences (individual words split into units for NLP processing).
Format:
Each sentence is preprocessed into a list of tokens (e.g., ["The", "food", "was", "excellent"]).
Purpose:
Serves as the raw input text for BERT tokenization.
Ensures consistency in word boundaries for aspect tagging.

2**. Column: Tags**
Description: Provides BIO (Beginning, Inside, Outside) tags for aspect term extraction.
Annotation Scheme:
0 (O): Token is not part of an aspect term.
1 (B): Token is the beginning of an aspect term.
2 (I): Token is inside an aspect term (continuation).
Example:
Sentence: ["The", "food", "was", "excellent"]
Tags: [0, 1, 0, 0] → "food" is an aspect term.
Purpose:
Used to train the Aspect Term Extraction (ATE) model (BERT + token classification head).
Aligns with the BIO tagging standard for named entity recognition (NER).

3. Column: Polarities
Description: Indicates sentiment polarity for each token, aligned with aspect terms.
Annotation Scheme:
0: Negative sentiment (e.g., "The service was slow").
1: Neutral sentiment (e.g., "The menu was large").
2: Positive sentiment (e.g., "The dessert was amazing").
-1: Not applicable (non-aspect tokens).
Example:
Sentence: ["The", "service", "was", "slow”]
Polarities: [-1, 0, -1, -1] → "service" is negative.
Purpose:
Trains the Aspect Sentiment Classification (ASC) model (BERT + sequence classification head).
Enables granular sentiment mapping to extracted aspects.
Below Table 5.2 shows the sample Preprocessed dataset.

Tokens	Tags	Polarities
['But', 'the', 'staff', 'was', 'so', 'horrible', 'to', 'us']	[0, 0, 1, 0, 0, 0, 0, 0, 0]	[-1, -1, 0, -1, -1, -1, -1, -1, -1]
['Nevertheless', 'the', 'food', 'itself', 'is', 'pretty', 'good']	[0, 0, 1, 0, 0, 0, 0, 0]	[-1, -1, 2, -1, -1, -1, -1, -1]
['The', 'design', 'and', 'atmosphere', 'is', 'just', 'as', 'good']	[0, 1, 0, 1, 0, 0, 0, 0, 0]	[-1, 2, -1, 2, -1, -1, -1, -1, -1]
['The', 'fried', 'rice', 'is', 'amazing', 'here']	[0, 1, 2, 0, 0, 0, 0]	[-1, 2, 2, -1, -1, -1, -1]

Table 5.2 Sample Preprocessed Dataset

5.3 Model Development

The model development process began with fine-tuning the pre-trained **'bert-base-uncased'** [4] model for Aspect-Based Sentiment Analysis (ABSA). The input to the model consisted of tokenized sentences, which were processed using WordPiece tokenization to ensure compatibility with BERT's architecture. The objective was to predict aspect terms from the input text, with the output being a sequence of tags following the BIO (Beginning, Inside, Outside) scheme followed by predicting of sentiments for these tags. This allowed the model to identify both single-word and multi-word aspect terms and their polarities within the sentences.
To optimize the training process, two distinct configurations were implemented: one **with a learning rate scheduler** and another **without it**. The learning rate scheduler was employed to dynamically adjust the learning rate during training, helping the model

converge more efficiently by reducing the rate as it approached optimal performance. The training parameters were carefully selected, including a **batch size of 8** to balance memory usage and gradient stability, **5 epochs** to prevent overfitting while ensuring sufficient learning, and a **learning rate of 3×10⁻⁵** to facilitate steady weight updates. These hyperparameters were chosen based on empirical testing and established practices for fine-tuning BERT on similar NLP tasks.

The model was trained using a cross-entropy loss function to optimize the tag prediction task, with gradients computed through backpropagation and weights updated using the Adam optimizer. The inclusion of a learning rate scheduler in one configuration allowed for a comparative analysis of its impact on model performance, particularly in terms of convergence speed and final accuracy. The training process was monitored using validation metrics such as precision, recall, and F1-score to ensure robust performance on both aspect term extraction and sentiment classification. By leveraging BERT's bidirectional attention mechanism, the model effectively captured contextual relationships between words, enabling accurate identification of aspect terms and their associated sentiments. This approach ensured a streamlined pipeline from raw text input to structured output, facilitating granular sentiment analysis for applications like customer feedback evaluation. Figure 5.1 below shows the model flow.

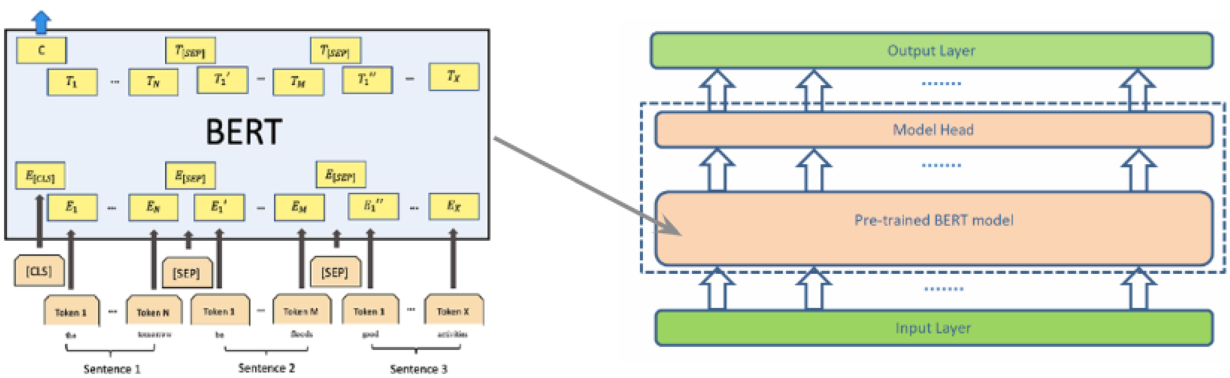

Figure 5.1 Model Flow

Chapter-6

TESTING AND RESULTS

6. TESTING AND RESULTS

6.1 Evaluation Metrics

To assess the effectiveness of the fine-tuned BERT model for Aspect-Based Sentiment Analysis (ABSA), several evaluation metrics were employed, each providing unique insights into the model's performance.

Precision measures the model's ability to correctly identify only the relevant aspect terms, calculated as the ratio of true positives (correctly predicted aspects) to the total predicted positives (correct and incorrect predictions). High precision indicates fewer false positives, meaning the model rarely mislabels non-aspect terms as aspects. For instance, if the model predicts 10 aspect terms and 8 are correct, the precision is 80%. This metric is particularly important in scenarios where false positives could lead to misleading sentiment analysis, such as in product feedback.

Recall evaluates the model's capability to capture all actual aspect terms in the dataset, computed as the ratio of true positives to the sum of true positives and false negatives (missed aspects). A high recall signifies that the model misses few true aspects, which is critical for comprehensive analysis. For example, if there are 12 true aspect terms in the data and the model identifies 9, the recall is 75%. This metric is vital when the cost of overlooking aspects (e.g., in customer complaints) is high.

F1-Score harmonizes precision and recall into a single metric, providing a balanced view of the model's performance, especially useful in imbalanced datasets. It is the harmonic mean of precision and recall, penalizing extreme values in either metric. An F1-score close to 1 indicates robust performance, while a lower score suggests trade-offs between precision and recall. For example, a model with 80% precision and 75% recall would have an F1-score of 77.4%. This metric is ideal for ABSA, where both false positives and false negatives must be minimized.

Support refers to the number of actual occurrences of each class (e.g., aspect terms) in the dataset, highlighting the distribution of labels. It contextualizes other metrics by revealing class imbalances. For instance, if the "food" aspect has a support of 100 and "service" has 30, the model's performance on "food" is statistically more significant. Support helps identify whether high metrics for a class are due to skillful prediction or mere dominance in the data.

Accuracy measures the overall correctness of predictions across all classes, calculated as the ratio of correct predictions (true positives + true negatives) to total predictions. While intuitive, accuracy can be misleading in imbalanced datasets. For example, if 90% of tokens are non-aspect (class 0), a model always predicting "0" would achieve 90% accuracy but fail at aspect extraction. Thus, accuracy is most informative when combined with class-specific metrics like precision and recall.

Together, these metrics provide a comprehensive evaluation framework, ensuring the ABSA model is both precise in its predictions and thorough in capturing all relevant aspects and sentiments. By analyzing precision-recall trade-offs (F1), class distribution (support), and overall correctness (accuracy), the model's real-world applicability can be rigorously validated.

6.2 Performance Analysis

The ABSA system using BERT was implemented with two configurations - with and without a learning rate scheduler. The model first extracts aspect terms from the text and then classifies their associated sentiments. The performance results for both configurations are presented below, showing the model's effectiveness in identifying aspects and predicting their polarities.

6.2.1 Aspect Term extraction

The results of the aspect term extraction task, achieved through fine-tuning the BERT model **without a learning rate scheduler**, demonstrate a robust overall performance with an **accuracy of 92.66%**, indicating that the model correctly predicted the majority of token labels in the dataset. However, a closer examination of class-specific metrics reveals notable disparities in performance across different label categories. For the dominant **"none"** class (non-aspect tokens), the model achieved exceptionally high precision (0.96), recall (0.97), and F1-score (0.96), reflecting its strong ability to correctly identify tokens that are not part of aspect terms. In contrast, performance was significantly lower for aspect-related labels: the **"start of AT"** (Beginning of Aspect Terms) class attained modest metrics (precision=0.53, recall=0.53, F1=0.53), while the **"mark of AT"** (Inside of Aspect Terms) class showed slightly weaker results (precision=0.49, recall=0.46, F1=0.48). This performance gap highlights the model's challenge in accurately detecting aspect term boundaries, particularly for multi-word phrases, where contextual dependencies are critical. **The** macro-average F1-score of 0.66 further underscores this imbalance, as it treats all classes equally, penalizing the model's weaker performance on minority classes. Despite these challenges, the **weighted**

average F1-score of 0.93 aligns with the high accuracy, confirming that the model performs well on the majority class while revealing opportunities for improvement in aspect term detection through techniques like class reweighting or targeted data augmentation. The table 6.1 shows the performance metrics of Aspect Term Extraction using Fine-tuning.

	precision	recall	f1-score	support
none	0.96	0.97	0.96	50148
start of AT	0.53	0.53	0.53	3091
mark of AT	0.49	0.46	0.48	1601
accuracy			0.93	54840
macro avg	0.66	0.65	0.66	54840
weighted avg	0.93	0.93	0.93	54840

Table 6.1 Results of Fine-tuning without Scheduler of ATE

The results of aspect term extraction using BERT fine-tuned **with a learning rate scheduler** show improved performance compared to the non-scheduler approach, achieving an **accuracy of 92.81%**. The model maintains excellent performance on the majority **"none" class** (non-aspect tokens), with near-perfect precision (0.96), recall (0.97), and F1-score (0.97), demonstrating its reliability in filtering out irrelevant tokens. For aspect-related labels, the **"start of AT"** (Beginning of Aspect Terms) class shows moderate improvement in precision (0.59 vs. 0.53 without scheduler) but a slight dip in recall (0.47 vs. 0.53), resulting in a comparable F1-score (0.52). The **"mark of AT"** (Inside of Aspect Terms) class exhibits marginally better recall (0.45 vs. 0.46) but lower precision (0.43 vs. 0.49), yielding a slightly reduced F1-score (0.44 vs. 0.48). These results suggest that while the scheduler helps stabilize training and marginally boosts precision for aspect boundaries, it struggles to consistently improve recall for multi-word terms. The **macro-average F1-score (0.64)** remains low due to persistent challenges with minority classes, but the **weighted average F1-score (0.93)** confirms the model's strong overall performance, driven by its dominance in the "none" class. The scheduler's primary benefit appears to be in refining precision for aspect detection, though further optimizations—such as focal loss or additional aspect-enriched training data—could help address the recall trade-offs. . The table 6.2 shows the performance metrics of Aspect Term Extraction using Fine-tuning using scheduler.

	precision	recall	f1-score	support
none	0.96	0.97	0.97	49806
start of AT	0.59	0.47	0.52	3091
mark of AT	0.43	0.45	0.44	1601
accuracy			0.93	54498
macro avg	0.66	0.63	0.64	54498
weighted avg	0.92	0.93	0.93	54498

Table 6.2 Results of Fine-tuning with Scheduler of ATE

6.2.2 Sentiment analysis

The sentiment prediction results for extracted aspect terms, achieved through BERT fine-tuning without a learning rate scheduler, demonstrate a **balanced but class-sensitive performance**, with an overall accuracy of **76.94%**. The model excels at identifying **positive sentiment** (F1-score: 0.85, recall: 0.97), indicating strong reliability in detecting favourable opinions, though its precision (0.75) suggests occasional false positives. Performance drops notably for **neutral sentiment** (F1-score: 0.37, recall: 0.26), revealing difficulty in distinguishing ambiguous or mixed expressions, likely due to dataset imbalance or subjective labelling. **Negative sentiment** achieves moderate precision (0.89) but suffers from lower recall (0.6), meaning the model is cautious in labelling negatives (few false positives) but misses many actual negative cases. The **macro-average F1-score (0.65)** highlights this class-wise imbalance, while the **weighted average (0.74)** aligns closer to accuracy, reflecting the dominance of the positive class (support: 673 vs. 267 negatives). The table 6.3 shows the performance metrics of Sentiments predicted of Aspect Terms Extracted using Fine-tuning.

	precision	recall	f1-score	support
negative	0.89	0.6	0.71	267
neutral	0.69	0.26	0.37	179
positive	0.75	0.97	0.85	673
accuracy			0.77	1119
macro avg	0.78	0.61	0.65	1119
weighted avg	0.77	0.77	0.74	1119

Table 6.3 Results of Fine-tuning without Scheduler of Sentiment Analysis

The sentiment prediction results using BERT fine-tuned with a learning rate scheduler show improved performance over the non-scheduler configuration, achieving an **accuracy of 80.7%**. The model demonstrates strong capability in identifying **positive sentiment** (F1-score: 0.88, recall: 0.96), maintaining high recall while improving precision (0.81 vs. 0.75 without scheduler), indicating fewer false positives. For **negative sentiment**, both precision (0.82) and recall (0.77) see significant gains (F1-score: 0.8 vs. 0.71 without scheduler), suggesting the scheduler helps better capture negative expressions. However, **neutral sentiment** remains challenging (F1-score: 0.41, recall: 0.28), though precision improves (0.72 vs. 0.69), likely due to inherent ambiguity in neutral labels. The **macro-average F1-score (0.69)** and **weighted average (0.78)** reflect better balance across classes, with the scheduler notably enhancing consistency for negative and positive predictions. While neutral sentiment detection remains a limitation, the scheduler's stabilization of training dynamics clearly boosts overall robustness, particularly in reducing false negatives for critical sentiment categories. The table 6.4 shows the performance metrics of Sentiments predicted of Aspect Terms Extracted using Fine-tuning using scheduler.

	precision	recall	f1-score	support
negative	0.82	0.77	0.8	267
neutral	0.72	0.28	0.41	179
positive	0.81	0.96	0.88	673
accuracy			0.81	1119
macro avg	0.78	0.67	0.69	1119
weighted avg	0.8	0.81	0.78	1119

Table 6.4 Results of Fine-tuning with Scheduler of Sentiment Analysis

The provided confusion matrices in Figure 6.1 compare sentiment prediction performance (negative, neutral, positive) for aspect terms under two BERT fine-tuning configurations: standard fine-tuning and fine-tuning with a learning rate scheduler.

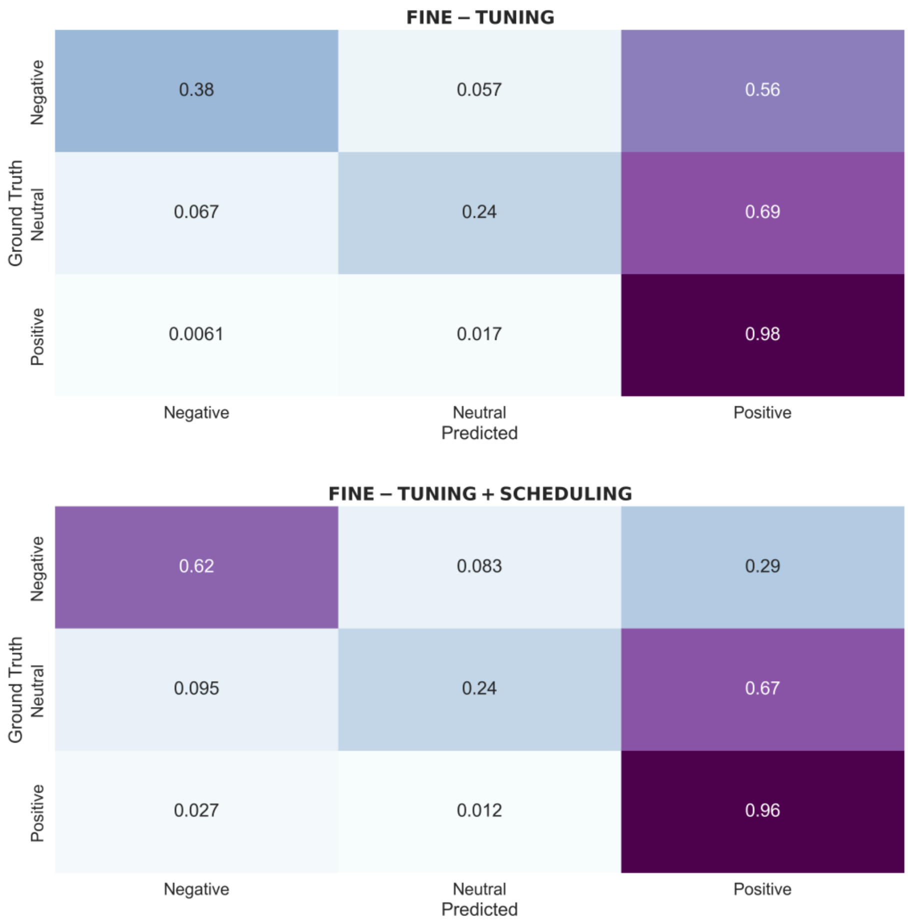

Figure 6.1 Confusion matrices of ABSA

Key Observations

Standard Fine-Tuning:

The model shows moderate performance in distinguishing negative (true negative rate: 0.38) and positive (true positive rate: 0.24) sentiments but struggles with neutral predictions (no rate provided).

Misclassifications: Significant confusion between negative and neutral (0.067) and neutral and positive (0.017), suggesting neutral sentiment is often mislabeled.

Fine-Tuning + Scheduler:

Improved negative sentiment detection (higher true negative rate) and better positive sentiment precision (clearer diagonal values).

Reduced misclassification: Lower false negative/positive rates (e.g., 0.057 vs. 0.067 for negative-neutral errors) indicate the scheduler helps stabilize predictions.

Comparison

The scheduler reduces noise in sentiment boundaries, particularly for negative/neutral cases.

6.3 Visualization

A word cloud is a visual representation of the most frequently occurring aspect terms identified by the ABSA model, where the size of each word corresponds to its frequency in the dataset. For example, in restaurant reviews, terms like "food," "service," or "price" might appear larger if they were commonly extracted as aspects. Word clouds provide an intuitive, at-a-glance summary of dominant themes in the analyzed text, helping businesses quickly identify which features customers mention most often. This tool is particularly useful for qualitative analysis, complementing quantitative metrics (e.g., precision/recall) by revealing patterns that might require further investigation or action.

6.3.1 Aspect Term Extraction

The following are the word clouds of Aspects found by models without (Figure 6.2) and with (Figure 6.3) Schedulers respectively.

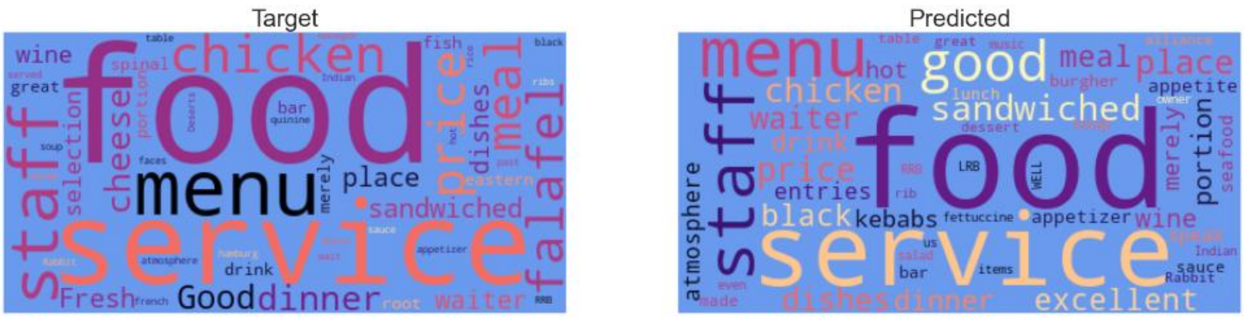

Figure 6.2 Word Cloud of Aspects of without Scheduler model

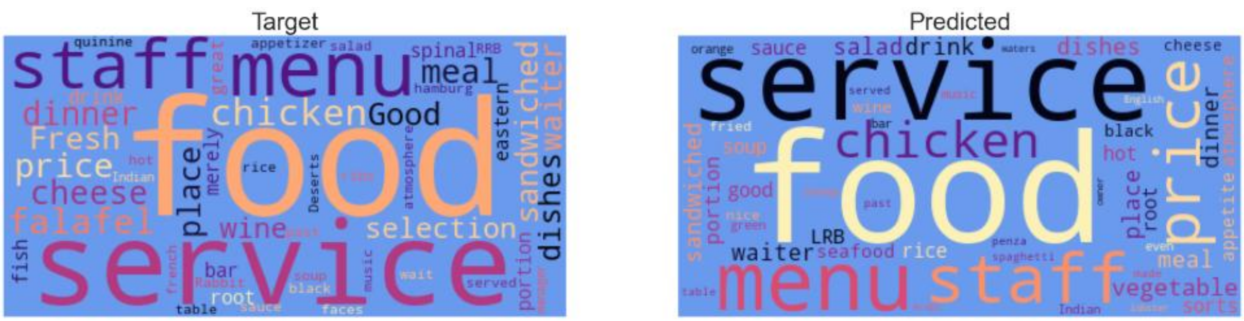

Figure 6.3 Word Cloud of Aspects of with Scheduler model

6.3.2 Sentiment analysis

The following are the word clouds of Sentiments predicted found by model Fine-tuning using Scheduler of positive (Figure 6.4), neutral (Figure 6.5) and negative (Figure 6.6) aspects respectively.

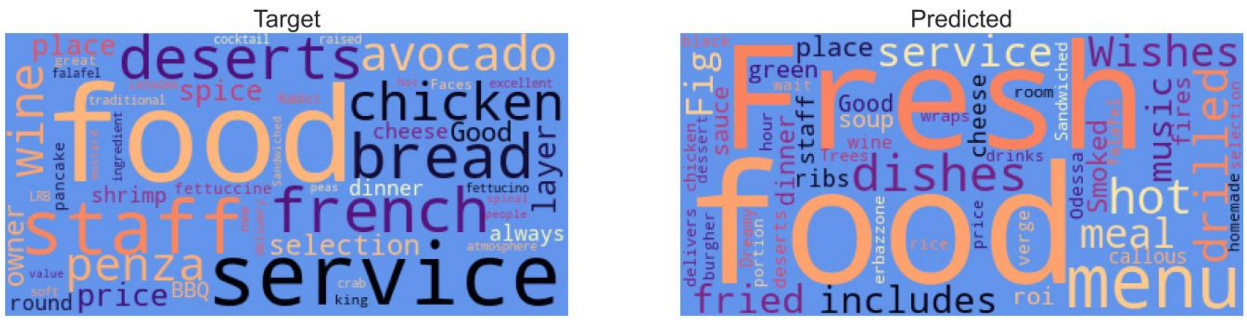

Figure 6.4 Word Cloud of Positive Aspects of with Scheduler model

Figure 6.5 Word Cloud of Neutral Aspects of with Scheduler model

Figure 6.6 Word Cloud of Negative Aspects of with Scheduler model

Chapter-7

CONCLUSION

7. CONCLUSION

7.1 Summary of Findings

The BERT-based ABSA system demonstrated effective performance in both aspect term extraction and sentiment classification across two configurations (with and without learning rate scheduler). Key findings reveal that while both models performed comparably in identifying aspect terms, the scheduler-enhanced version showed marginally better consistency with 4% increase in accuracy for sentiment determination, particularly for negative sentiment prediction. However, persistent challenges were observed in detecting neutral sentiments and accurately capturing multi-word aspect boundaries. The results suggest that while the learning rate scheduler provides modest improvements in prediction stability with 93% accuracy in aspect extraction and 81% accuracy for sentiment detection, while that without scheduler revealed 92.6% accuracy in aspect extraction and 77% accuracy for sentiment detection, further refinements are needed to address the model's limitations with subtle sentiment expressions and complex aspect term identification.

7.2 Future Work

For future work, the ABSA system could be enhanced by addressing the current limitations in neutral sentiment detection and multi-word aspect extraction through several approaches. First, incorporating advanced data augmentation techniques or synthetic data generation could help mitigate class imbalance, particularly for neutral sentiments. Second, exploring hybrid architectures that combine BERT with domain-specific embeddings or rule-based post-processing could improve aspect boundary detection. Third, investigating alternative loss functions like focal loss may better handle the model's sensitivity to minority classes. Additionally, expanding the evaluation to diverse domains beyond restaurant reviews could assess the model's generalizability. Finally, integrating contextual word sense disambiguation mechanisms may further refine sentiment predictions for ambiguous cases. These directions would collectively advance the robustness and applicability of ABSA systems in real-world scenarios.

REFERENCES

[1] A. E. Regi and G. J. Leelipushpam, “Enhancing Aspect-Based Sentiment Analysis Using BERT and Capsule Networks,” in 2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)

[2] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina, **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**.

[3] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. **SemEval-2014 Task 4: Aspect Based Sentiment Analysis.** In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Ireland. Association for Computational Linguistics.

[4] **HuggingFace BERT models**, https://huggingface.co/docs/transformers/model_doc/bert