FACULDADE
DE ECONOMIA
UNIVERSIDADE
DO PORTO

Deep Learning in Sentiment Analysis: A comparative analysis between traditional models and BERT.

Inês Isabel Moreira da Silva

Dissertation

Master in Modelling, Data Analysis and Decision Support Systems

Supervised by:

PhD Bruno Veloso

2025

Acknowledgments

Completing this dissertation has been a significant academic and personal achievement. I would like to take this opportunity to thank everyone who helped me along the way, whether directly or indirectly.

Firstly, I would like to thank the University of Porto for providing me with the opportunity to learn from excellent professors, especially my supervisor, Bruno Veloso, who provided invaluable guidance and support throughout the dissertation process. His advice and helpful feedback have shaped this dissertation from the outset.

I would also like to thank my family for their unwavering support throughout this process and throughout my entire life. Thank you to my parents and brothers for keeping me grounded by believing in me, even when I doubted myself.

Finally, I would like to thank all my friends for their understanding, humour and patience, which made the difficult days easier.

Abstract

Sentimental analysis has gained importance in natural language processing, with applications that span from monitoring consumer opinion to corporate and public policy decision-making. Its origins date back to opinion research in the early 20th century, and with the emergence of blogs, social media, and online reviews in the mid-2000s, it has regained its popularity. Since then, sentiment analysis has progressed beyond manual keyword counting and now includes machine learning classifiers, including deep learning models like LSTM, CNN, and transformer-based architectures like BERT.

However, there are still many obstacles to overcome, even with these developments. Traditional approaches often struggle with analyzing complex linguistic occurrences, such as sarcasm, denial, and the existence of multiple sentiments in a single statement. Additionally, the lack of high-quality annotated datasets for Aspect-Based Sentiment Analysis (ABSA) makes it harder for models to generalize across languages and domains. Therefore, two research questions are the main focus of this dissertation. Firstly, it takes into account the performance of deep learning architectures such as CNN, LSTM, and BERT in sentiment analysis tasks. Secondly, it explores how BERT can be adjusted to handle ABSA more effectively, specifically with regards to multi-aspect sentiments, sarcasm, and negation.

A comparative experimental framework was created using several benchmark datasets in order to address these problems. The fine-tuning of BERT was done using a pre-trained model while CNN and LSTM were trained from scratch. The evaluation metrics used to measure performance included accuracy, precision, recall, F1-score, macro-F1, weighted-F1. The findings were statistically compared using the Autorank framework, since it ensures a solid and reasonable comparison across models while providing insights into the unique advantages and disadvantages of each architecture.

The results show that BERT consistently performs well across datasets, balancing precision and recall across sentiment classifications, despite its higher computational costs. CNN is beneficial when speed is crucial, but LSTM is a strong substitute in resource-constrained situations, offering a reasonable balance between accuracy and efficiency. Moreover, applying BERT to the SemEval dataset for ABSA showed that it can handle multiple-aspect sentiments like sarcasm and negation, which highlights its suitability for fine-grained sentiment analysis tasks.

Keywords: Aspect-Based Sentiment Analysis, Neural Networks, LSTM, CNN, BERT

Resumo

A análise de sentimentos tem vindo a ganhar importância no processamento de linguagem natural, com aplicações que vão desde a monitorização da opinião dos consumidores até à tomada de decisões empresariais e políticas. As suas origens remontam à investigação de opinião no início do século XX e, com o aparecimento de blogues, redes sociais e avaliações online em meados da década de 2000, recuperou a sua popularidade. Desde então, a análise de sentimentos progrediu para além da contagem manual de palavras-chave e inclui agora classificadores de aprendizagem automática, incluindo modelos de deep learning como LSTM, CNN e arquiteturas baseadas em transformadores como o BERT.

No entanto, mesmo com estes desenvolvimentos, ainda há muitos obstáculos a ultrapassar. As abordagens tradicionais debatem-se frequentemente com a análise de ocorrências linguísticas complexas, como o sarcasmo, a negação e a existência de vários sentimentos numa única frase. Além disso, a falta de conjuntos de dados anotados de alta qualidade para a Análise de Sentimentos Baseada em Aspectos (ABSA) dificulta a generalização dos modelos entre diferentes línguas e domínios. Assim, duas questões de investigação são o foco principal desta dissertação. Inicialmente, tem em conta o desempenho de modelos de deep learning como o CNN, LSTM e BERT em tarefas de análise de sentimentos. Em segundo lugar, explora como o BERT pode ser ajustado para lidar com o ABSA de forma mais eficaz, especificamente no que diz respeito a sentimentos multi aspeto, sarcasmo e negação.

Para resolver isto, foi criado uma estrutura experimental comparativa utilizando vários conjuntos de dados de referência. O fine-tuning do BERT foi efetuado utilizando um modelo pré-treinado, enquanto a CNN e o LSTM foram treinados de raiz. As métricas de avaliação utilizadas para medir o desempenho incluíram accuracy, precision, recall, F1-score, macro-F1, weighted-F1. Os resultados foram comparados estatisticamente utilizando a estrutura Autorank, uma vez que esta assegura uma comparação sólida e razoável entre modelos, ao mesmo tempo que fornece informações sobre as vantagens e desvantagens únicas de cada.

Os resultados mostram que o BERT tem um desempenho consistentemente bom em todos os conjuntos de dados, equilibrando a precision e recall nas classificações de sentimentos, apesar dos seus custos computacionais mais elevados. A CNN é benéfica quando a velocidade é crucial, mas a LSTM é um forte substituto em situações de recursos limitados, oferecendo um equilíbrio razoável entre acurácia e eficiência. Além disso, a aplicação do BERT ao conjunto de dados SemEval para a ABSA mostrou que pode lidar com sentimentos de múltiplos aspetos, como o sarcasmo e a negação, o que realça a sua adequação a tarefas de análise de sentimentos de granularidade fina.

Palavras-chave: Análise de sentimento baseada em aspetos, redes neurais, LSTM, CNN, BERT.

Contents

Glossary

ABSA - Aspect-Based Sentiment Analysis
AE-LSTM - Aspect Embedding Long Short-Term Memory
AF-LSTM - Aspect-Fusion Long Short-Term Memory
ATAE-LSTM - Attention-based Long Short-Term Memory with Aspect Embedding
AT-LSTM - Attention-based Long Short-Term Memory
BERT - Bidirectional Encoder Representations from Transformers
BERT-ASC - BERT-based Auxiliary Sentence Construction
BERT-PT - BERT Post-Training
BERT-SPC - BERT Sentence Pair Classification
Bi-LSTM - Bidirectional Long Short-Term Memory
BoW - Bag-of-words
CEF - Constant Error Flow
CG-BERT - Context-Guided BERT
CNN - Convolutional Neural Networks
CRISP-DM - Cross Industry Standard Process for Data Mining
CV - Cross Validation
DeBERTa - Decoding-enhanced BERT with Disentangled Attention
DL- Deep Learning
DRA - Dynamic Re-weighting Adapter
DR-BERT - Dynamic Re-weighting BERT
FN - False Negative
FP - False Positive
GRU - Gated Recurrent Units
H-LSTM - Hierarchical Long Short-Term Memory
H-SUM - Hierarchical aggregation
LOOCV - Leave-One-Out Cross Validation
LPOCV - Leave-Pair-Out Cross-Validation
L-LDA - Labeled Latent Dirichlet Allocation
LSTM - Long Short-Term Memory
ML - Machine Learning
MLM - Masked Language Modeling
NLP - Natural Language Processing
NSP - Next Sentence Prediction
OOV - Out-of-Vocabulary
P-SUM - Parallel aggregation
QACG-BERT - Quasi-Attention Context Guided
ReLu - Rectified Linear Unit

RNN - Recurrent Neural Networks
ROC AUC- Receiver Operating Characteristic Area Under the Curve
TC-LSTM - Target-Connection Long Short-Term Memory
TD-LSTM - Target-Dependent Long Short-Term Memory
TF-IDF - Term Frequency Inverse Document Frequency
TN - True Negative
TP - True Positive

List of Figures

List of Tables

Chapter 1

Introduction

The motivation for this study and the problem being investigated are discussed in this chapter. The aims, objectives, risks, and limitations that were taken into account are outlined in it. Finally, the overall structure of the dissertation is described.

1.1 Motivation

In the present age of digitalization, social media and e-commerce platforms have become tools for companies to be closer to their customers. These channels are often used by customers to express their opinions and sentiments, resulting in a vast amount of data that can be analyzed to improve decision-making.

In this way, the analysis of customer feedback about the products and services is crucial for any business to thrive. The development of informed and personalized marketing strategies that are tailored to the customer's needs is possible through the knowledge of their behavior. By analyzing this data, the company gains a competitive advantage, as it is then able to meet the customers' needs and increase the brand reputation.

Nevertheless, sentiment analysis can be applied to multiple sectors beyond business and marketing, namely finance, health, government, and others. By monitoring the sentiment expressed on social media and in newspapers within a given sector, it is possible to gain insights into the stock market and monitor the risk. This can also be applied to the health sector, where it can be used to track the propagation of diseases and access public health trends. Furthermore, it can be used to analyze public opinion on government policies, political campaigns, and other social issues. (Islam et al., 2024)

Sentiment analysis, also known as opinion mining, plays a major role in the process of identifying, extracting and organizing sentiment from text, audio or visual data. This is achieved through the application of Natural Language Processing (NLP) techniques, which are employed to analyze textual data and identify the text polarity (positive, negative or neutral). (Al-Qablan, Mohd Noor, Al-Betar, & Khader, 2023)

Informal language, sarcasm, negation and multiple aspect sentiments are some of the difficulties that arise when dealing with data collected from the internet. For that reason, traditional approaches to sentiment analysis based on NLP and Machine Learning (ML) algorithms may face some limitations in capturing complex sentiments, affecting the model's performance.

This dissertation is motivated by the recognition of the significant impact that deep learn-

ing techniques have had on the field of sentiment analysis. The use of these techniques has enabled the autonomous learning of features in large datasets, which is essential for the revolutionary changes in the field. (Y. Wu, Jin, Shi, Liang, & Zhan, 2024)

1.2 Problem Description

Despite the advancements in deep learning for NLP, ABSA is still a difficult task. In addition to needing large, labeled datasets, traditional models like CNN and LSTM have trouble with sarcasm, subtle or mixed sentiments, and domain transferability. Although transformer-based models, such as BERT, provide improvements, they still have drawbacks in terms of interpretability, generalization across domains, and handling implicit elements like multi-aspect sentiment and negation. Furthermore, it is challenging to determine which architecture performs best in various conditions because there are no systematic comparative studies.

Project aim and goals

The principal aim of this dissertation is to train various deep learning models and perform a comparative analysis of their performance. To accomplish this, two different models will be trained on five different datasets: the Amazon Product Review, IMDB, Yelp review, SemEval-2014 Task 4 and T4sa datasets. The models selected for training are the Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN). Furthermore, pre-trained Bidirectional Encoder Representations from Transformers (BERT) were employed across all datasets for sentiment classification, with the SemEval dataset additionally adjusted for ABSA.

Therefore, the specific objectives of this work are:

- Gain a comprehensive understanding of the current state of the art in deep learning for sentiment analysis, as well as the various deep learning architectures that will be used.
- Analyze and gain practical experience in training and testing different deep-learning techniques for text data.
- Implement Aspect-Based Sentiment Analysis (ABSA) using pre-trained BERT models.
- Conduct a comparative analysis evaluation of the performance of the deep learning models in each dataset.
- Examine the limitations of each deep learning model.

Risks and Possible Limitations

- Detection of complex sentiments, such as sarcasm or other ambiguous expressions.
- High computational power to train models like BERT.

1.3 Dissertation Structure

The theory of sentiment and aspect sentiment analysis is introduced in chapter two and the basic concepts, techniques, and deep learning models are summarized. This contributes usefully to this study by setting the necessary background for understanding the research context.

The next chapter reviews existing literature and highlights current advances in aspect-based sentiment analysis concerning LSTM, CNN, and BERT. The objective of this chapter is to find gaps in current research and to highlight the need for the approaches discussed in this dissertation.

Chapter four provides a comprehensive definition of the study methodology that follows the CRISP-DM model framework. Understanding the business (4.1) and the data (4.2), preparing the data (4.3), modeling (4.4), and evaluation (4.5) are the five phases of the methodology chapter. This method sets up a comprehensive theoretical foundation and procedural framework to guide the implementation of the entire research process (Data Science Process Alliance, 2024).

The methodology chapter provides detailed explanations of the business objectives, data collection and exploration procedures, dataset specifications, and preprocessing techniques for LSTM, CNN and BERT models. Additionally, it describes the modeling approaches to be used and the evaluation metrics and procedures to be employed throughout the study.

In the following chapter, practical implementation and experimental results are presented. The established methodology is demonstrated through experiments with different deep learning architectures for sentiment analysis. The experiments involve comparing performance, implementing the auto-rank framework, analyzing the confusion matrix, adapting BERT for ABSA, and presenting the main empirical findings.

Finally, chapter six ends by summarizing the experimental results, discussing the overall findings, identifying future research directions, and offering final remarks on the research contributions and limitations.

Chapter 2

Theoretical Foundations

The theoretical basis supporting the study is presented in this chapter. An introduction to sentiment analysis is given, which includes text pre-processing, feature extraction, classification, evaluation, and validation. Furtermore, ABSA is introduced and neural network approaches are explored, including RNN, LSTM, GRU, CNN, and BERT.

2.1 Introduction to Sentiment Analysis

As mentioned before, Sentiment Analysis relies on NLP techniques to analyze textual data and then determine the emotional tone behind it (Al-Qablan et al., 2023). Identifying the emotional tone behind a text involves four main steps: Text Processing; Feature identification; Classification; Evaluation and Validation.

2.1.1 Text Processing

The text-processing phase, which is also referred to as the pre-processing phase, involves the cleaning and preparation of text data for classification. Normally, the data that is retrieved from social media or reviews is unstructured, making this a required step to reduce the text noise and dimension, leading to faster and less prone models to overfit.

According to Jin, Cheng, Wang, and Cai (2023), the steps of text-processing are as follows:

- **Tokenization**, which implies the division of the text into smaller units called tokens. The tokens can be individual words, punctuation, numbers, and other elements depending on the application and language.
- **Removing irrelevant or redundant words**, as they do not carry inherent sentiment information. This includes stop words such as "the", "and", "so" "if"; punctuation; special characters such as emojis and hashtags.
- **Lowercase all text**, ensuring that the same words that appear with different capitalizations are treated in the same way.
- **Treatment of negations**, which plays a crucial role in sentiment analysis, as it can completely reverse the polarity of a sentence. After identifying the negation word ("no", "never", "ever", etc.), there is the need to adjust the following adjectives and adverbs to express the true felling. However, there is the need to also determine the scope of

the negation to determine how far into the sentence the negation applies, which can be challenging to develop algorithms and rules that take that nuances into account.

- **Stemming and Lemmatization**, which involves the reduction of the words to their radical to simplify the analysis and reduce the data dimensionality. While Stemming focuses on reducing the word, not worrying if the words will be valid, Lemmatization focuses on finding the base of the word (or lemma) taking into account the context and grammatical class, generating only valid words. As evident, lemmatization is a relatively more computationally complex process than stemming.

In summary, when looking at the bigger picture, it is evident that the text processing phase contributes to the minimization of noise in the data, improving the model's performance and accelerating the classification process.

2.1.2 Feature Extraction

In the subsequent stage, it is vital to convert raw text into numerical representations that can be utilized by machine and deep learning algorithms (Jin et al., 2023). Here, the important information of the text, that indicates the polarity and intensity of the sentiment, is captured.In this way, Jin et al. (2023), described the following feature extraction and representation techniques.

- **Bag-of-words (BoW)**, considers the frequency of each word in a document, building a document-word-matrix. In fact, it treats the text purely as a collection of words and overlooks the order or grammatical structure (Jin et al., 2023).

- **Term Frequency Inverse Document Frequency (TF-IDF)** is an enhancement of the BoW model that weights the importance (TF) of a word in a document against its rarity in the whole text (IDF) (Jin et al., 2023).

- **Word embeddings**

  This method of representation allows us to represent words as vectors in muti-dimensional space. Here, the distance and direction between the vectors corresponds to the similarities and relationships between them. Examples of representation techniques include Word2Vec, GloVe and Fast Text; altough these come with meaningful limitations (for instance, difficulty in distinguishing words with different meanings and the absence of contextual understanding).

- **Contextual embeddings** (BERT, GPT)

  The contextualized embeddings emerge from the sophisticated deep-learning fundamentals that consider the context of the whole phrase, improving performance and representations, particularly in handling polysemy and context-dependent meanings. Nevertheless, they require more computational resources compared to word embedding, which are more suited for computing simpler tasks

2.1.3 Classification

The polarity of sentiments in text is ultimately determined through the classification phase. According to Mao, Liu, and Zhang (2024), sentiment classification approaches include sentiment lexicon and rule-based approaches, traditional machine learning and deep learning approaches.

- **Sentiment Lexicon and rule-based approaches**

Sentiment Lexicon and rule-based approaches use pre-constructed sentiment dictionaries and a set of rules to determine the text polarity. It's the most simple and intuitive classification approach.

The dictionaries, in simple terms, contain words and/or phrases with their associated scores, which indicate whether they are positive, negative, or neutral. Then, after all the preprocessing and feature identification, the algorithm looks for the score of each word/sentence in the dictionary and the overall sentiment is computed by combining these scores.

However, here it's necessary to add some rules to deal with more complex sentiments such as negations. Otherwise, the presence of negative words can distort the polarity of sentiments and lead us to wrong conclusions.

- **Machine learning Approaches**

Machine learning (ML) is the science of guiding computers to carry out tasks without precise commands. In here algorithms are trained to process big amounts of data to recognize patterns and predict outcomes for new situations.

According to Al-Qablan et al. (2023), there are three types of ML approaches: the supervised, the unsupervised and the semi-supervised.

Supervised approaches are widely used in sentiment analysis. They require labelled data, divided into training and test sets. The model learns from the labeled data and predicts the sentiment of new data. Popular algorithms include Naive Bayes, Support Vector Machines, Logistic Regression , Decision Trees and Random Forest (Al-Qablan et al., 2023).

Unsupervised approaches do not need labeled data. They are used for tasks such as clustering, where the algorithm identifies patterns in the data. K-means clustering is a common technique in unsupervised sentiment analysis (Al-Qablan et al., 2023).

Lastly, Semi-supervised ML combines labeled and unlabeled data. They are useful when little labeled data is available. One example is the Recursive Neural Network (Al-Qablan et al., 2023).

However, there are some drawbacks to using ML for sentiment analysis. The acquisition of labeled data for these algorithms can be costly and time-consuming. Special training is necessary for them because people express emotions in different ways in different situations. Otherwise, their accuracy will decrease. They also have difficulties in capturing complex feelings like sarcasm, irony and contextual nuances, because ML models can't always identify the scope of the negation and adjust the sentiment polarity accordingly. Lastly, ML models are unable to effectively track how opinions change over time, potentially missing important shifts.

- **Deep learning Approaches**

Deep learning depends on structures that mimic brain neuron structures named neural networks and address complex tasks that usually involve human-like thinking (IBM, 2024b). They tend to surpass some of the limitations of classical machine learning, but this ability can sometimes be costly in the computations and demand even more labeled data (Al-Qablan et al., 2023).

The popular deep-learning approaches employed in sentiment analysis are CNNs, RNNs, LSTM, Bi-LSTM, and Transformer-based models, all discussed in the chapters that follow.

2.1.4 Evaluation and Validation

Sentiment analysis ends by assessing the effectiveness of the model used for the classification of emotional tone in texts. Evaluation and validation are now key components towards ensuring that the model is reliable, robust, and effective in predicting sentiment from text data.

The first part involves the comparison of model predictions with the real sentiment labels given to the data for evaluation. The evaluation metrics below are a measure of the ability of the model to classify a sentiment indicated in data.

- **Confusion Matrix:** counts the number of data correctly and incorrectly classified. True Positive (TP) and True Negative (TN) signify examples where the model precisely predicts the positive and negative classes. In reverse, False Negative (FN) and False Positive (FP) refer precisely to the examples whose model wrongly sees as negative and positive, respectively.

- **Accuracy** is calculated based on the percentage of occurrences that were predicted to be true from the total number of occurrences. The computation is done as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.1}$$

- **Precision** measures the accuracy of positive sentiment predictions, which is the percentage of true positive predictions among all positive predictions. It is provided by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.2}$$

- **Recall or Sensitivity** is a measure of the model's capacity to identify all true positive sentiments. It is calculated as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.3}$$

- **Specificity** uses the same reasoning as sensitivity but concentrates on negative cases. It examines the model's capacity to identify all actual negative cases and its calculated using the following mathematical expression::

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.4}$$

- **F1 score** reflects the balance between precision and recall, being useful when the class distribution is imbalanced. The calculation method is as follows:

$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.5}$$

Therefore, evaluation is essential to point out the specific weaknesses of the model's performance, suggesting areas where the models or training data can be improved. In terms of validation, these methods ensure that the model doesn't overfit to data and can generalize well in new data. According to Gama, Carvalho, Faceli, Lorena, and Oliveira (2017), common validation methods include:

- **Hold-Out Validation**

The Hold-Out approach entails splitting the original dataset into training (70%) and test sets (30%). Training takes place in the training set and the model's performance is evaluated in the test set, enabling us to observe the model´s performance on unseen data (Gama et al., 2017).

- **Cross validation (CV)**

CV involves the splitting of datasets in a randomized or stratified way. The most common approach is the K-fold CV, where the data is split into k subsets, with each one of them serving as validation data.

According to Alhamid (2020), there are two types of variation of CV techniques which are Leave-One-Out Cross Validation (LOOCV) and Leave-Pair-Out Cross-Validation (LPOCV). The first situation involves k being equal to the number of data points in the dataset. During each iteration, only one observation is utilized for validation while the others are utilized for training. This process is repeated until all data points are used as validation data. LPOCV is based on the same reasoning, but instead of single data points, pairs of data points are used.

Nonetheless, validation can battle with class imbalance, noise and mislabeled data in test sets, leading to biased performance estimates and influence the reliability of the evaluation.

2.2 Aspect-Based Sentiment Analysis

The subarea of sentiment analysis known as Aspect-based Sentiment Analysis (ABSA) is responsible for identifying and extracting the polarity of sentiment associated with specific aspects or attributes within the text (Yadav, 2020).

ABSA is a finer approach to sentiment analysis and other NLP tasks like as Aspect Extraction, Sentiment Classification, and Opinion Summarization (Hua, Denny, Wicker, & Taskova, 2024). As stated by Hua et al. (2024), ABSA has the following subtasks:

- **Aspect Extraction**: It's the process of extracting some elements of interest from the given text. In a phone review, the elements could be 'price', 'camera', 'performance', etc.

- **Opinion Extraction** involves the search for words, phrases, or statements that express opinions about the identified aspects to discover the sentiment behind them. For example, the aspect camera could be referred to as 'fantastic' and / or 'user friendly'.

- **Aspect Sentiment Classification** determines whether a particular aspect is positive, negative, or neutral. This is a comprehensive understanding of how people perceive specific traits or characteristics in the text, not just the overall sentiment.

- **Aspect of Category Detection** categorizes similar aspects into predefined categories, such as 'Hardware'.

- **Aspect Category Sentiment Analysis** evaluates the overall sentiment associated with each category.

The complex nature of natural language and sentiment nuances presents several complex challenges in ABSA.

- **Domain and Context Dependency**, in which there is the need to train this system to adapt to domain-specific datasets, enabling the understanding of these terms (Mao et al., 2024).

- **Extracting Implicit Aspects**, which requires contextual inferences since it's possible to make a negative comment without using negative words.

- **Co-reference resolution**, when several words or sentences refer to the same entity.

- **Detecting sarcasm**

- **Multilingual support**, which includes the development of systems that work in multiple languages. This is challenging due to linguistic and cultural differences.

Despite these challenges, ABSA has several advantages such as ability to understand customer preferences by analyzing sentiments towards specific aspects. Consequently, it helps companies understand what the strengths and weaknesses of their products or services are, aiding in decision-making of improvement strategies (Hua et al., 2024).

The future of ABSA lies in the use of pre-trained models to improve the model's accuracy and efficiency, transfer learning to improve the speed of the development of ABSA systems, and real-time sentiment analysis tracking, to enable timely and effective decisions.

2.3 Neural Networks in Sentiment Analysis

The human brain inspires Neural Networks, which are algorithms consisting of interconnected layers of artificial neurons that process and transmit information. Usually, these algorithms have at least three layers: an input layer, one or more hidden layers, and an output layer. Neural Networks' capability in sentiment analysis is attributed to their skill in learning complex representations and modeling relationships between words.

2.3.1 Recurrent Neural Network

A Recurrent Neural Network (RNN) is a form of artificial neural network designed to effectively handle sequential or time-series data. RNNs have feedback loops to retain information from prior computational steps. In essence, at each step in time, the output of the

network is affected not by the present input alone but also by incorporating the information retained from prior inputs, so it can form dependencies over time (Das, Tariq, Santos, Kantareddy, & Banerjee, 2023).

According to Hochreiter and Schmidhuber (1997), the "memory" of prior computational steps allows RNNs to deal with tasks that imply temporal dependencies such as speech processing, language modeling, machine translations, musical composition, etc. In simpler terms, it allows the recognition of the order of words and prediction of the next word in a sentence, understanding the context of a sentence to make a correct translation, generating coherent sequences of musical notes, and making decisions in dynamic systems that depend on history.

Despite that, RNNs have some limitations with regard to handling long-range dependencies, primarily due to gradient fading during training. Gradients shrink exponentially when they are propagated backward through many stages, which is why this issue occurs. Long-term interactions are given less weight in the model than short-term interactions, making it challenging to capture dependencies from earlier in the sequence (Colliot, 2023).

2.3.2 Long Short-Term Memory (LSTM)

As stated by Hochreiter and Schmidhuber (1997), the ability to introduce memory cells is what allows LSTM to overcome the gradient fading's limitations. Important information can be kept flowing through the network without fading away because of the Constant Error Flow (CEF) mechanism in these memory cells. To control the information on each cell, three gates (input, forget and output gates) filter what the model remembers and forgets, enabling only the useful information to be kept. The algorithm can concentrate on complex relationships thanks to the combination of these mechanisms.

Additionally, there is an extension of LSTM that processes the input sequence in both directions (forwards and backwards), named Bidirectional LSTM. This allows the model to capture contextual information from both sides of a word or phrase, leading to more informative representations (Al-Qablan et al., 2023).

2.3.3 Gated Recurrent Units (GRUs)

GRUs are a more simplified approach to LSTM that enables us to address the problem of gradient fading in a faster and less computationally expensive manner. The regulation of memory cell information involves the introduction of only two gates (update and reset gates), which determine the amount of past information to forget and new information to keep (Singh, 2024). Therefore, LSTM and GRU algorithms can store and retrieve information for longer periods of time, making them effective at capturing long-term dependencies in text data. In recent studies, LSTM achieved better performance than GRUs in terms of word error rate, but the execution time was longer (Shewalkar, Nyavanandi, & Ludwig, 2019).

2.3.4 Convolutional Neural Networks (CNNs)

CNNs have been very successful at carrying out NLP tasks and they have proven to be highly effective at recognizing images, audio and speech. This algorithm architecture is generally comprised of six major components as per Diwan and Tembhurne (2022).

1. **Input Layer**

The input layer is able to receive input that is either an image, text, or other structured data. The input for textual data can consist of a sequence of word or character embeddings. When it comes to images, the input is often a three-dimensional matrix that includes height, width, and color.

2. **Convolutional Layers**

   Convolutional Layers involve converting the imput layer into numerical representations. Informative filters that take into account local features of the text input (such as n-grams and word embeddings) are used by CNNs to combine higher-level representations. In each convolutional layer, a variety of features are captured using multiple filters (also known as kernels). Each filter is trained to identify a specific pattern in the input, like n-grams and phrases in text. Then, the convolutional layer creates feature maps that represent the filter's response to the input (IBM, 2024a)

3. **Pooling layers**

   Regarding pooling layers, the model's robustness and efficiency are improved by reducing the dimension of feature maps created in the convolutional layers, even if some information is lost (IBM, 2024a).

4. **Activation layers**

   In order to learn complex relationships between features, activation layers introduce non-linearity into the model. The most common activation functions are Rectified Linear Unit (ReLU) and Sigmoid.

5. **Fully Connected Layers**

   Fully Connected Layers are used to perform the final prediction tasks, which utilize the features extracted from previous layers and filters.

6. **Output Layer**

   The Output Layer is responsible for producing the final output of the model. The result of classification problems is a class, and typically, the softmax function is employed to generate a probability distribution for the possible classes.

To sum up, as the layers of CNNs are crossed, the complexity of the network increases, enabling the identification of larger components of the input. In the initial layers, relatively simple aspects are identified. As the data proceeds through the layers of the CNN, increasingly larger components are recognized.

As a consequence, these algorithms are highly effective in capturing local patterns and spatial relationships in text data. Despite this, CNNs' performance is a result of their data quality, network architecture, and hyperparameters.

As stated by Diwan and Tembhurne (2022), CNNs offer several advantages, including: Automatically acquiring and interpreting context-relevant features; Computational efficiency; Noise robustness, especially in handling grammatical errors and abbreviations commonly seen in informal text; Improved generalization, especially when trained on large data sets, and improved performance on the model by exploiting different pre-trained embeddings; and, lastly, multimodal sources can be supported since textual information can be inputted to achieve better prediction.

2.3.5 Bidirectional Encoder Representations from Transformers (BERT)

BERT is a type of NLP model that was developed by Google in 2018. This model was pre-trained on a large corpus text like Wikipedia, using language modeling tasks such as:

- **Masked Language Modeling (MLM)**, where some words/tokens are masked in a sentence for training the model to predict the words based on the context around each one of them.

- **Next Sentence Prediction (NSP)**, where the model tries to predict whether one sentence logically follows another, aiding the improvement of tasks such as question answering and text classification.

These two tasks are essential for the model to learn the contextual representation of words, aiding the model's understanding of the meaning of them in different contexts and helping the model to learn inter-sentence relationships.

Therefore, BERT is used to process language in a way that is similar to how humans think. This new unsupervised model processes text in both directions via transfer learning and has substantially transformed sentiment analysis (Chakkarwar, Tamane, & Thombre, 2023).

The transformer model is the basis of this model's architecture, which includes multiple layers of attention mechanisms and feed-forward neural networks (Su, 2024). In a nutshell, it enables learning from both left and right context in a sentence about a specific word without the need for a large-scale annotated text.

However, BERT merely provides a general comprehension of the language, which needs to be adjusted for specific tasks. Fine-tuning comes into play here. To put it simply, during fine-tuning, the model begins with the knowledge it acquired during pre-training. Then, a smaller dataset that is specific to the NLP task is used for further training and to produce outputs that are tailored to the task. To improve performance on new tasks, the pre-trained weights are updated and adjusted during model training on specific tasks.

Fine-tuning presents benefits such as better performance on specific tasks, and less data needed since the model already has a strong language foundation and it is faster to train. Therefore, with this bi-directionality, BERT sets itself apart from the conventional models by capturing meanings in much greater depths (Chakkarwar et al., 2023). However, training such powerful algorithms requires massive computation capacity, which could result in slower inference, making it difficult for real-time applications.

Chapter 3

Literature Review

The primary topic of discussion in this chapter is the exploration of the previous and current deep learning approaches that have been proposed for ABSA. The purpose of this is to identify any potential gaps in the literature.

3.1 Aspect-Based Sentiment Analysis with Long-Short Term Memory

In terms of ABSA, the order of the words in a sentence is considered by LSTM when processing sequential data in time steps, while capturing dependencies and relationships between words. This helps to improve the understanding of the overall context related to an aspect. During aspect extraction, this algorithm can identify elements in a sentence that are indicative of aspects by learning its patterns and context clues. After the identification of aspects, LSTM classifies how certain words correlate (positively, negatively, or neutrally). Then, the emotions are encoded in the models allowing them to effectively analyze the different sentiments associated with the various aspects.

However, the standard LSTM models may not be suitable for aspect-dependent sentiment analysis since they treat all the words in the same way regardless of the aspect being evaluated (Tang, Qin, Feng, & Liu, 2015). This is challenging since different aspects in the same sentence can have different sentiment polarities. For those reasons, a considerable number of studies have explored the use of LSTM for ABSA, with promising results.

From 2015 to 2016, Tang et al. suggested two advanced algorithms for LSTM in ABSA, namely Target-Dependent LSTM (TD-LSTM) and Target-Connection LSTM (TC-LSTM). According to this author, these algorithms are better than traditional ML methods for identifying the feelings behind an aspect in a text. Here, TD-LSTM uses a dual-LSTM architecture to study the feelings expressed towards a specific target within the text. As a result, the process involves looking at the context before and after each target separately and categorizing the sentiment by combining hidden vectors to form a representation of the sentence.

On the other hand, TC-LSTM has shown advancements in sentiment classification performance when compared to the standard LSTM and TD-LSTM. This algorithm incorporates the semantic connection between target words and their surrounding context. In simpler terms, at each position, it processes two inputs simultaneously: contextual word embeddings and a target vector (derived from averaging target word vectors) (Tang, Qin, & Liu, 2016).

As a result, capturing the meaning of contextual words is a difficult task for both algorithms. TD-LSTM considers all context words equally and uses the same sentence representation regardless of the target words. In contrast, TC-LSTM employs the target-context connection to evaluate this, but it takes more time to train because there are more parameters (Tang et al., 2016).

Moreover, Hierarchical LSTM (H-LSTM) uses a BI-LSTM in the sentence and evaluation layers, allowing to "leverage both intra- and inter-sentence relations" (Ruder, Ghaffari, & Breslin, 2016a, p. 1). This approach enables the algorithm to understand the structure of sentences and their context within the whole text. The main advantage of this approach is their multilingual applications since they're able to capture hierarchical relationships in text without requiring linguistic preprocessing or specific adaptations.

H-LSTM's dependence on sentence structure and training data presents limitations, particularly when working with low-resource languages, specific domains, or limited training data. Furthermore, it doesn't significantly improve LSTM results, presents difficulties when dealing with complex sentences, and the computational complexity tends to increase due to the high parameter count. (Ruder et al., 2016a; Wang, Huang, Zhu, & Zhao, 2016)

Following that, Attention-based LSTM (AT-LSTM) was proposed by Wang et al. (2016). By using this algorithm, it is possible to selectively focus on textual elements that are related to the target aspect. The evaluation of semantic relationships is done by calculating importance weights for context words and using a neural network. To create unique vector representations for different aspects during training, AE-LSTM included specific aspect embeddings. The improvements make it possible to have a better semantic understanding of the aspects themselves (Wang et al., 2016).

Attention-based LSTM with Aspect Embedding (ATAE-LSTM) is the most promising model developed by Wang et al. (2016). This algorithm combines the AT-LSTM and AE-LSTM, since it includes aspect information from the earliest stages of LSTM and maintains the context through the network.

Moreover, Tay, Tuan, and Hui (2018) also proposed Aspect-Fusion LSTM (AF-LSTM) that uses a different approach. This algorithm uses specific layers to model the relationships between aspects and context words, while including correlation operators to model word-aspect relationships. To achieve an accurate ABSA, AF-LSTM models the associative relationship between aspects and words to understand the importance of them.

The LSTM architectures mentioned earlier struggle with balancing model complexity and performance gains, as more complex models, such as embeddings, fusion, or attention, can sometimes result in poorer results. Moreover, architectures where parameters are expanded (such as AE-LSTM and ATAE-LSTM) result in increased computational complexity and hence increased training difficulties. Finally, neither of these variables completely solves the challenge of capturing complex interactions between aspects and different parts of a phrase, especially in more nuanced cases where they may not be able to differentiate sentiments (Tay et al., 2018; Wang et al., 2016).

In summary, LSTM plays a major role in capturing long-range relationships between words in a sentence, while dealing with the sequential nature of the text and may include attention mechanisms to focus on the most relevant parts of a sentence for a given aspect. Additionally, it can be combined with other neural networks (such as CNNs) to extract more complex features.

3.2 Aspect-Based Sentiment Analysis with Convolutional Neural Networks

CNNs are often used in ABSA for aspect extraction and sentiment polarity classification. Aspect Extraction is treated as a multi-label classification problem, where the model is trained to produce probability distributions over the aspects. Here, a threshold is used to discard features with lower probabilities and in some cases the models can be adapted to predict multiple aspects per phrase.

Schmitt, Steinheber, Schreiber, and Roth (2018) saw performance improvements when they used an end-to-end trainable neural network to model both tasks of aspect extraction and sentiment polarity classification. In this case, the output of the model is a vector that predicts the presence, absence, and polarity of each aspect.

The CNN-T model, which is an adaptation of CNN for ABSA, is capable of making multiple class predictions per input, as stated by Mulyo and Widyantoro (2018). The training process for this algorithm is the same as standard CNN; however, CNN-T utilizes a threshold to categorize outputs. This threshold is specific to each attribute, and it is determined by analyzing the training results, taking into account the trade-off between correctly and incorrectly classified data.

As reported by Mulyo and Widyantoro (2018), the challenges of CNN-T are related to finding an optimal threshold. This is an important but difficult task, as thresholds may vary between different datasets and they may not work well with the data. Additionally, it requires a large amount of training data, and the performance tends to drop when dealing with informal language, ambiguities and complex sentences.

Although CNN algorithms are already successful in ABSA, they are often integrated with other deep learning models to enhance the performance of these tasks. For instance, since CNNs can extract local features and spatial patterns, they can feed them to LSTM or BI-LSTM models to leverage long term dependencies in sequences (Cahyadi & Khodra, 2018). This approach allows CNN to process texts while BI-LSTM handles document context. Other adaptations involve implementing end-to-end architectures that merge feature extraction and polarity classification, enhanced by attention mechanisms for better token relevance weighting (Schmitt et al., 2018).

Furthermore, the effectiveness of ABSA can be improved by using unlabeled data, as it can facilitate word embedding training (Ruder, Ghaffari, & Breslin, 2016b). GloVe, Word2Vec and FastText are examples of the different types of word embeddings used in CNN models for ABSA.

As stated by Mulyo and Widyantoro (2018); Ruder et al. (2016b); Schmitt et al. (2018), GloVe is a word embedding model that has been trained on large datasets; however, it is characterized by the absence of character-level information. Furthermore, Word2Vec is successful in ABSA due to skip-gram and Continuous BoW methods that predict word relantionships, but it presents some challenges when dealing with morphologically rich languages. Finally, FastTest is more flexible due to the application of subword information, which allows it to deal with unseen words and perform better in complex morphological languages.

Training these embeddings on either domain-specific or domain-independent data is possible, depending on their availability. Consequently, this approach produces word representations that are more robust and generalized, which helps to expand semantic knowledge in new contexts.

The main obstacle lies in dealing with words that were not present during training, known as Out-of-Vocabulary (OOV). Techniques such as GloVe and Word2Vec either ignore them or replace them with an 'unknown' token. FastText was a more advanced approach because it breaks words into subwords to learn representations of those subpieces instead of treating them as a single unit. This makes it possible to generate meaningful embeddings and particularly excels by combining with CNN models for ABSA tasks.

3.3 Aspect-Based Sentiment Analysis with BERT

As mentioned before, the common approach for ABSA is fine tuning BERT, where it has a significant importance in the adaptation of the model to the desired data and task (Ahmed et al., 2022). Fine-tuning for ABSA tasks involve approaches such as sentence pair classification, auxiliary sentence construction and cross-domain data training.

Some studies have demonstrated performance improvement when using phrase pair classification models. In here, sentence pair classification tasks are created by structured input pairs, with original text and aspect query, allowing to focus on specific aspects. Therefore, some adaptations of BERT to ABSA were developed such as BERT-PT (Post-Training) and BERT-SPC (Sentence Pair Classification). As stated by Zhang et al. (2022), BERT-PT uses additional review data during training to make the model more aware of sentiment expression expressed in reviews, while BERT-SPC pairs text with specific aspect queries.

Nevertheless, pair classification models present some limitations such as focusing on overall sentiment rather than aspect-specific regional semantics. For fine tuning, it requires a significant amount of labeled data. Also, finding implicit aspects, modeling text-aspect relationships, changing when different aspects are taken into account, and dealing with multiple aspect phrases are challenging. Lastly, some approaches do not fully exploit BERT's potential since it is only used as an embedding layer (Ahmed et al., 2022; Z. Wu & Ong, 2021; Zhang et al., 2022)

In order to overcome these limitations BERT-based Auxiliary Sentence Construction (BERT-ASC) was proposed by Ahmed et al. (2022). BERT-ASC addresses the difficulty of mapping implicit aspects to their indicators. According to these authors, this model uses Labeled Latent Dirichlet Allocation (L-LDA), where aspect specific vocabularies are created by pairing each aspect with semantically related words. The similarity between the seed words and the words in the target review are measured, in order to see how the text relates to each aspect. As a result, auxiliary sentences to represent implicit aspects are created, allowing the model to learn aspect-specific representations.

Also, several studies modify BERT through models with context-guided attention. A study conducted by Z. Wu and Ong (2021), demonstrated that Context-Guided BERT (CG-BERT) uses attention mechanisms that consider a wider context when analyzing sentiment. An improvement of this model was achieved with Quasi-Attention Context Guided (QACG-BERT), which learns the importance of the context, focusing on specific parts of text by learning what can be ignored (Z. Wu & Ong, 2021).

According to the authors, the limitation of CG-BERT and QACG-BERT is the complexity of computation and interpretation. This is a result of the complex initialization and fine tuning requirements, particularly QACG-BERT with complex attention mechanisms. Furthermore, their reliance on context quality and labeled data presents challenges in addressing implicit aspects and cross-domain generalization.

An end-to-end framework was developed to integrate multiple ABSA steps without the need for separate modules for each step (Li, Bing, Zhang, & Lam, 2019). These frameworks are seen as a sequence labeling problem, where it processes raw input text and automatically provides a detailed sentiment analysis on the specific aspects mentioned in text. Then, in aspect extraction the specific aspects mentioned in text are identified by using pre-trained models such as BERT, attention models or NLP techniques (Karimi, Rossi, & Prati, 2020).

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) improves performance over standard BERT models, by processing separately position (syntax) and content (semantics) information within a text (Marcacini & Silva, 2021). A study followed by Zhang et al. (2022), introduced DR-BERT (Dynamic Re-weighting BERT), which shows an enhancement of ABSA by adding a Dynamic Re-weighting adapter (DRA) after each layer of BERT. While BERT focuses on the general understanding of the sentence, DRA focuses on specific areas of a sentence, reweighting the most important words.

DeBERTa and DR-BERT have similar computational complexity challenges. While DeBERTa utilizes disentangled attention mechanisms, DR-BERT adds complexity by adding an attention mechanism called DRA. Generalizing to new domains is a struggle for both models and they require significant labeled data and fine-tuning for achieving better performances. Also, it is challenging for them to identify implicit aspects in text and their decisions are complicated due to complex mechanisms. Lastly, when dealing with long-term dependencies, they face the challenge of complex relationships between aspects and sentiments (Ahmed et al., 2022; Karimi et al., 2020).

To improve the performance of BERT models in ABSA tasks, Parallel aggregation (P-SUM) and hierarchical aggregation (H-SUM) were explored by Karimi et al. (2020). These algorithms explore the hidden layers of BERT, allowing deeper semantic representations of the input sequences. While P-SUM processes BERTs final layers independently and in parallel, where each layer gets its own prediction path, with the loss calculated in the end. H-SUM creates a hierarchical structure, where the information of the hidden layers is combined with the previous ones. The first one excels in capturing independent layer-specific features, while the other one is better at handling complex feature interaction across layers.

As highlighted by Karimi et al. (2020) and Ahmed et al. (2022), the key limitations of models like P-SUM and H-SUM are similar to those of DeBERTa and DR-BERT when considering computational complexity, requirement of fine tuning and labeled data, difficulty in domain generalization, interpretability and handling implicit aspects. However, the limitations of P-SUM and H-SUM are from processing and aggregation of multiple BERT layers, with focus on layer aggregation and loss combination.

In summary, modern architectures like CG-BERT, DR-BERT, and DeBERTa propose various advanced techniques like context-guided attention, the method of dynamic reweighting, and disentangled attention to handle aspects that are implicit in the text, multiple aspects, or out-of-domain scenarios.

3.4 Future in Sentiment Analysis

Integrating sentiment expressed in different modes like text, images, audio, and video can help to gain a more comprehensive approximation of the sentiment. In this way, the future of sentiment analysis involves exploring more advanced models to combine information from a range of different types of content. Furthermore, it is necessary to have models that can

capture the context of various modalities such as the relationship between facial expressions in video and the sentiment expressed in the corresponding text (Islam et al., 2024).

While DL architectures such as BERT are promising in the understanding of context of languages, the future research involves the development of even more sophisticated models that can capture subtle nuances of language and deal with complex sentences with multiple clauses and dependencies. Additionally, there is also the need to leverage the pre-trained models on large data sets to improve performance in specific domains and low-data tasks (Mao et al., 2024).

Nowadays, ethical and privacy issues have emerged as one of the main concerns of the application of sentiment analysis. In this way, there is the necessity to utilize advanced techniques for anonymization of the data, allowing the protection of the identity of the individuals along with the collection of the data (Islam et al., 2024).

ABSA is making significant improvements in its ability to extract information from text, becoming more proficient at identifying implicit and multi-word aspects. While doing so, it is creating more elaborate models that reveal the connections between these aspects and the emotions they communicate, including modifiers and negations (Mao et al., 2024).

Then, one of the most difficult problems for sentiment analysis is the detection of sarcasm and irony. Therefore, there is the need for advanced understanding of subtle clues given in the context. At the moment, two main areas of research are using the user´s history and knowledge of the subject and developing multi-task learning, which combines sarcasm detection with other relevant tasks, such as emotion detection (Wnkhade, Rao, & Kulkarni, 2022).

To handle the constant evolution of informal language, we need to plan strong strategies. Slang, shortened words, and emojis are all part of this. A few recent accomplishments include systems that are capable of comprehending the significance of emojis in various scenarios and, to improve the accuracy of online analysis of real-world conversations, informal language lexicons are constantly updated (Mao et al., 2024).

3.5 Discussion

With respect to LSTM, TD-LSTM and TC-LSTM depend on averaging basic word embeddings, which may fail to capture multi-word targets effectively. Although hierarchical and attention-based models show promise, they are not transferable across domains because they depend on external sources and are not applicable to low-resource languages. Furthermore, a lot of techniques prioritize one sentence at a time, which could result in overlooking any disclosure structure that may be reflected in the sentiment of other sentences.

In order to overcome error propagation in CNNs (tendency to commit more errors once the first one is made) the method of joint modelling is applied. As a result, tasks such as aspect detection and sentiment classification are modeled into a single algorithm. This results in a better capture of dependencies between aspects and related sentiments. Nevertheless, this requires a large amount of labeled and high-quality data, and the application to real word cases is severely impacted by the lack of datasets for specific languages and/or domains,.

The generalization of CNN across different domains and languages is normally restricted by limited labeled data. Moreover, the capacity of adaptation to different datasets is also subject to constraints imposed by predefined feature extraction methods, such as thresholds or rule based approaches.

In summary, the application of LSTM and CNN to ABSA is encountering challenges that are not fully comprehended. For example, by struggling with sarcasm detection, capturing subtle nuances, analyzing mixed sentiments in a sentence, dealing with imbalanced datasets, and consequently leading to bias towards frequent sentiment classes.

There are some gaps in the literature review of BERT applied to ABSA, where some topics not fully covered in the field such as lack of robust models for detecting implied aspects and handling complex sentiment. In addition, there is also the difficulty of applying existing methods to cross-domain scenarios.

Finally, there is a lack of integration of external knowledge when fine-tuning BERT and a lack of interpretability. Even though end-to-end training allows us to train and handle different ABSA tasks in one single model, challenges regarding the balance between flexibility and fine-grained control over individual aspects are shown.

This study contributes to the field by addressing challenges in sentiment analysis and ABSA that are still unresolved in the literature. In order to fill these gaps, this study applies ABSA specifically to the SemEval 2014 dataset (one of the few publicly accessible datasets created for aspect-level evaluation) and compares LSTM, CNN, and BERT in a methodical manner across several datasets. Although ABSA analysis is limited to one dataset, it illustrates how BERT can be modified for complex sentiment tasks and provides valuable insight into model performance trade-offs.

In conclusion, the field of ABSA needs better unsupervised implicit aspect detection methods, which are more efficient, less computationally intensive and have improved interpretability. Here, the models need to combine different techniques to meet the challenge of natural language variability and multi-aspect relationships in long-range dependencies. Also, to improve real-world applicability, consideration should be given to expanding the variety by incorporating greater linguistic and domain diversity, as the language of these models is mostly in English and limited to product or service reviews.

Chapter 4

Methodology

This chapter provides a detailed explanation of the steps that were taken in the experimental phase, including the framework utilized, as well as the main tools and software utilized to address the research question.

The methodology that is going to be employed in this dissertation is the Cross Industry Standard Process for Data Mining (CRISP-DM). According to Schröer, Kruse, and Gómez (2021), CRISP-DM is a widely recognized framework in data science that consists of the following main phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment

4.1 Business Understanding

The initial phase of CRISP-DM is Business Understanding, which is the fundamental phase for settling the objectives and goals of the project. Additionally, this phase involves determining the availability of resources, accessing project requirements and evaluating potential risks and contingencies. Subsequently, the project plan is developed including the selection of technologies and tools and the formation of detailed plans for each project phase (Data Science Process Alliance, 2024; Schröer et al., 2021).

The primary aim of this study is to train and assess the performance of LSTM and CNN models in various sentiment analysis datasets, while also employing pre-trained BERT models. In addition, the SemEval dataset will be used to investigate the effectiveness of BERT in ABSA. The comprehensive theoretical foundation of sentiment analysis and, in particular aspect-based sentiment analysis, for the different deep learning methods selected was done in the chapter regarding theoretical foundations and literature review.

Then, for gaining practice experience in training deep learning models, the primary software tools that will be used are Python and Jupyter Notebook to provide an interactive environment for developing and executing the selected models. However, in the case of a lack of computational resources, Google Colab will be used as an alternative, as it provides cloud-based processing that ensures the efficient execution of the models. Additionally, in section A.1, it can be found a Gant Chart that summarizes the project timeline.

Business Motivation

In today's society, companies have a strong dependence on customer feedback to guide various functions, such as marketing strategies, product improvement, and development.

Therefore, it is necessary to have automated methods for analyzing thousands of reviews in a manner that is less time-consuming and less prone to bias. However, simply identifying whether a review is neutral, negative, or positive is insufficient. The companies must comprehend which features are receiving praise and criticism in order to keep improving their goods and services.

Nevertheless, it is still quite difficult to apply deep learning models to ABSA. While LSTMs and their variations have trouble capturing multi-word targets, sarcasm, implicit elements, and nuanced or mixed attitudes within a single sentence, CNNs frequently suffer from error propagation and limited generalization across domains, as noted in the literature. In this way, these flaws can cause predictions to be skewed, particularly when datasets are unbalanced. Even with BERT's significant improvements in managing contextual dependencies, current methods struggle with interpretability, cross-domain adaptability, and the recognition of inferred characteristics. In addition, companies need reliable and effective ABSA systems to extract detailed information from customer feedback, resulting in a disconnect between academic advancement and practical application. Thus, the objective of this study is to evaluate the relative performance of CNN, LSTM, and BERT architectures while examining how BERT could be more capable of handling ABSA issues and providing more useful sentiment insights.

4.1.1 Project Objectives and Measures of Success

As stated earlier, this project is focused on developing deep learning models that can extract useful insights from reviews, with a particular emphasis on ABSA to enhance prediction accuracy. By categorizing specific product aspects that lead to customer dissatisfaction, companies can prioritize feature improvements and manage brand reputation

Additionally, the initial stage of CRISP-DM requires the establishment of clear success criteria. In this manner, it is possible to evaluate the project outcomes and measure the effectiveness of different deep learning models when performing sentiment analysis in specific domains. Therefore, the primary success criteria are:

1. **Comparative performance evaluation**
   The study aims to quantify the relative performance differences between the LSTM, CNN and BERT models, and to assess BERT's effectiveness in ABSA using the SemEval dataset. Therefore, the study does not aim to define the models effectiveness by reaching a particular performance threshold. Furthermore, the aim is to use the models to address research questions and provide significant evidence on the suitability of different deep learning models for sentiment analysis.

2. **Reproducibility and Robustness**
   Fair comparison should be ensured through consistent pre-processing, training parameters and evaluation metrics. To measure success, the models will be evaluated against standard NLP metrics such as accuracy, precision, recall, F1-score, training time, and inference time.

3. **Computational Efficiency analysis**
   The three deep learning architectures computational demands, such as training time and inference speed, will be taken into account during this analysis to assess the trade-off between model efficiency and predictive performance.

4.1.2 Constraints

The model performance and project delivery can be impacted by the following key constraints:

1. **Data Limitations**
   Class imbalance is present in some datasets, with an uneven distribution across sentiment categories. Therefore, training can be impacted by this imbalance and lead to biased predictions toward majority classes. In this way, it has the potential to influence the reliability of sentiment analysis results for business decision-making.

2. **Temporal limitations**
   A shortened timeline may restrict the project's ability to experiment with models and optimize hyperparameters. In order to get around this restriction, model development tasks must be carefully prioritized, with an emphasis on strategies that are most likely to produce insightful data.

3. **Computational resources limitations**
   Model training resources were identified as a potential bottleneck, which was the main constraint on the development of this dissertation. By strategically using cloud-based GPU resources through Google Colab, this constraint has been lowered, allowing the training of complex deep learning models within project goals.

4.2 Data Understanding

After understanding the main goals and objectives of the project, there is a need to understand the data (Schröer et al., 2021). This phase includes the selection of datasets and performing exploratory data analysis to gain insight into data relationships and quality.

In this case, Amazon Product Review, IMDB, Yelp reviews, SemEval-2014 Task 4 and T4sa datasets were chosen to perform sentiment analysis. Additionally, in the following chapters, exploratory data analysis is going to be performed to address the sentiment labels and distribution of aspects.

4.2.1 Data Collection

One of the project's primary objectives is to develop and evaluate deep learning models across diverse domains to assess their generalizability. Therefore, datasets were collected from reputable sources including Hugging Face Hub, Kaggle, and domain-specific platforms in order to accomplish this, and covering the following application areas:

1. **Consumer products and cosmetics:** Amazon products reviews within the "Beauty and Cosmetics" category that provide customer feedback on several products and brand experiences.

2. **Movies:** IMDB movies reviews dataset, enabling the extraction of user sentiments towards entertainment content.

3. **Local Businesses:** Yelp customer reviews and ratings for various service businesses.

4. **Social Media:** T4SA dataset includes tweets for sentiment analysis, which allows for informal and real-time analysis of sentiment expressions.

5. **Services:** SemEval 2014 task 4 restaurant reviews includes aspect-based sentiment annotations, which allows for a more granular sentiment analysis. Additionally, out of the datasets used in this study, only the SemEval 2014 dataset provides the annotations necessary for ABSA, making it the only dataset suitable for these experiments.

In this way, deep learning models can be evaluated for their generalizability and performance in different sentiment tasks and domains, including text characteristics, domain vocabulary, and patterns of sentiment expressions.

Table 4.1: Dataset description

Dataset	Size	Features	Labels	Missing Values	Domain
IMDB	50 000	Text and label	Binary (0,1)	None	Movie Reviews
Amazon	701 528	10 columns including rating, title, text and metadata	Rating scale (1-5)	None	Product Reviews
T4sa	1 179 957	TWID, text and sentiment	3-point scale (0=neg, 1=neutral, 2=pos)	None	Social media content
Yelp	700 000	Text and label	Rating scale (0-4)	None	Business Reviews
SemEval	4 720	Sentence, Aspect Term and Polarity	Categorical (positive, negative, neutral and conflict)	1 021	Aspect-based reviews

Table 4.1. shows that the datasets have a significant variation size. The T4sa dataset represents the largest dataset in this collection, with 1 179 957 social media content. Following that, Yelp and Amazon datasets are also relatively large, making them suitable for deep learning approaches that take advantage of large training datasets. The SemEval dataset's limited size, with 4 720 samples, presents a distinct challenge, which necessitates models to achieve strong performance with minimal training data.

Different approaches to representing sentiment are employed in the datasets, which reflect the diverse nature of the underlying domains and analytical goals. A binary classification system that simplifies sentiment into positive or negative categories is used by the IMDB dataset. In contrast, the Amazon Reviews dataset uses a five-point rating scale (1-5) to capture subtle levels of satisfaction. Similarly, the Yelp dataset utilizes a five-point scale representing values from 0 to 4. The T4SA dataset has a three-point scale that is associated with negative,

neutral, and positive sentiments. Lastly, SemEval differentiates itself from other datasets by identifying specific aspect terms in sentences and assigning polarity labels (positive, negative, neutral or conflict) to them.

4.2.2 Data Exploration

Analyzing the distribution of target variables is an essential part of the data understanding phase following the CRISP-DM framework. The sentiment distribution visualizations provide valuable insights into class balance and potential modeling considerations for each dataset.

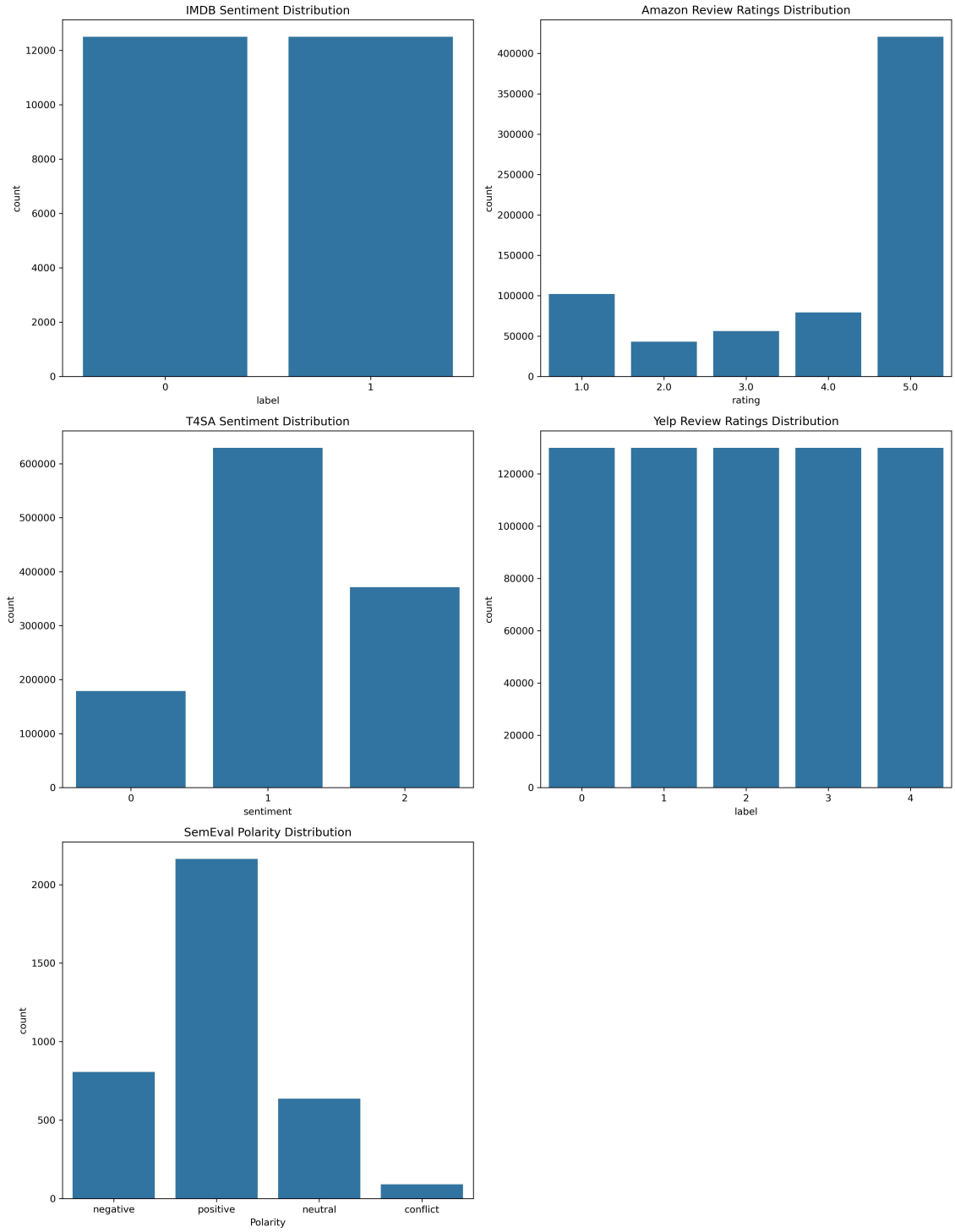

Figure 4.1: Data Exploration: Distribution of classes for each dataset

The IMDB and Yelp sentiment distribution charts indicate well-balanced datasets with equal representation of negative and positive reviews. Classification tasks can benefit from this balanced distribution as it prevents bias towards any particular class and provides equal learning opportunities for both sentiment polarities.

According to the Amazon Reviews dataset, the distribution is highly skewed towards the highest rating (5.0). This represents a significant bias towards positive sentiment, with relatively few instances in the lower rating categories. In the modeling phase, special attention needs to be given to avoid bias by the classifier towards the majority class due to the substantial imbalance in positive reviews. Similarly, there is a significant discrepancy in the SemEval dataset's class distribution, with positive sentiment samples outnumbering the other three classes (negative, neutral, and conflict).

The T4SA tweets sentiment distribution shows that neutral sentiment is the most prevalent at approximately 620.000 instances, with the negative sentiment being the least common. The distribution of social media content in this dataset indicates that it tends to have neutral expressions, with positive content being more prevalent than negative content. It is important to address the significant class imbalance during model development.

Textual Characteristics of the Dataset

Table 4.2: Word, Sentence, and Text Statistics

Dataset	Word			Sentence			Text		
	Min	**Mean**	**Max**	**Min**	**Mean**	**Max**	**Min**	**Mean**	**Max**
IMDB	3	4	6	11	283	2 818	52	1 325	13 704
Amazon	0	4	52	0	38	3 098	0	173	14 989
T4sa	0	4	35	4	20	139	28	108	160
Yelp	0	4	37	1	155	1 213	1	732	5 637
SemEval	0	4	10	2	18	79	5	88	357

The multi-dimensional text analysis in table 4.2 reveals significant differences between the various datasets in terms of text, sentence, and word granularities.

IMDB reviews are the most challenging in terms of text-level patterns, averaging 1325 words and some reaching 13 704 words. This exceeds BERT's 512-token limit and may necessitate truncation strategies. Unlike other reviews, Amazon's are extremely brief, consisting of an average of only 173 words, and some even have zero characters. This could indicate data quality issues that require filtering and validation steps before model training. The typical constraints of Twitter are reflected in T4SA, where texts average 108 words in length. While Yelp reviews tend to be 732 words long, there are still reviews that exceed the maximum token limit of 5 637 words.

In addition, sentence level analysis can reveal complementary patterns that helps to explain the complexity present in the reviews. There are multiple sentences in IMDB's reviews (mean of 283 and a maximum of 2818), which suggests that hierarchical processing approaches, that first analyze individual sentences and then combine them for document-level sentiment analysis, may be advantageous for the data. In contrast, SemEval is distinguished by its lean sentence structure (mean: 18), which is the right choice for ABSA that focuses on opinions.

Lastly, granularity at the word level provides crucial information for tokenization and vocabulary strategies. The average word length of these datasets ranges from 4.06 to 4.52 characters, suggesting similar linguistic complexity, but the maximum word lengths can reveal some dataset-specific difficulties. In addition, specified tokenization for hashtags, URLs, or technical terms can be needed in the Amazon and T4SA datasets.

By comprehending these text length patterns, it is possible to select the correct preprocessing methods, efficiently allocate computing resources, and design appropriate evaluation approaches for optimal sentiment analysis. To perform sentiment analysis optimally, it is necessary to meet the unique processing requirements created by the interaction between text, sentence, and word-level characteristics.

Word Frequency Analysis

The language patterns that are typical of movie reviews are revealed by the word frequency analysis of the **IMDb** dataset (Figure A.2). As expected, the word 'movie' is the most frequently used, followed by adjectives like 'good', 'great' and 'bad', which reflect the high level of opinion expressed in these reviews. According to the analysis, reviewers place a lot of emphasis on storytelling and production quality, frequently referring to terms like 'story', 'character' and 'film'. Furthermore, the frequent use of engagement words like 'like', 'watch' and 'see' indicates how much these reviews are focused on sharing experiences and giving recommendations to other viewers.

The language used in the **Amazon** review analysis is heavily focused on products and user experiences. As shown in Figure A.3, the most common words are those related to 'product', 'hair', and 'skin'. The use of emotional words like 'love', 'great' and 'good' by reviewers suggests they are trying to recommend products to others. The difference between Amazon reviews and movie reviews is that customers focus on practical things like price, color, and delivery time. To discuss how well products perform, they use terms like 'use,' 'work,' and 'quality'.

The **T4SA** tweet analysis demonstrates how diverse social media languages are, as illustrated in Figure A.4. Positive words like 'love', 'happy', 'great' and 'amazing' are frequently used by users, which could indicate greater level of enthusiasm in their posts. Furthermore, the way people tweet about current events and immediate experiences is reflected in words like 'new', 'today' and 'day'.

Observing **Yelp** reviews (Figure A.5), it is clear that people prioritize food, service, and their dining experiences. Common words like 'food', 'good' and 'place' show this is restaurant content. Reviews tend to give recommendations and frequently use words like 'ordered', 'time' and 'back' to describe service details when discussing how things worked. The fact that Yelp reviews often mention the physical experience of visiting a restaurant and interacting with staff and other customers makes it stand out from other review.

According to the **SemEval** dataset (Figure A.6), people prioritize both food quality and customer experience due to frequent mentions of 'food' and 'service' in reviews. Specific food words like 'wine', 'menu' and 'chicken' are also used by people when describing their meals. In addition, practical matters like 'tables', 'staff' and 'waiting' are discussed, which demonstrate the significance of good service for a satisfying restaurant experience.

Quality Assessment

It can be observed in table 4.1. that the majority of datasets do not have missing values in critical columns. The lack of duplicates in the T4SA and Yelp datasets makes them almost

analysis ready with minimal data preprocessing requirements.

The majority of datasets are complete and do not contain any missing values. However, the SemEval dataset has significant gaps, with 1 021 samples (21.6%) not having annotations for both aspect term and polarity. Furthermore, there are 29 rows that are duplicates in this dataset. It is expected that these missing values will be valid due to the fact that not all reviews have specific aspect terms that can be evaluated for sentiment. Despite this, this dataset is exceptionally valuable for developing models that can identify opinion targets and their associated sentiments in text, leading to more detailed and actionable insights.

Moderate duplication issues can be found in the IMDB dataset (96 in train and 199 in test sets), whereas the Amazon dataset is more challenging with 7 275 duplicate rows. To prevent bias in model training and evaluation, it is important to carefully handle these duplications during the data preparation phase.

In summary, the preprocessing will be directed towards developing appropriate strategies to deal with duplicates in IMDB and Amazon datasets, while determining the most effective way to deal with structured missing values in SemEval. The T4SA and Yelp datasets are able to proceed without intervention.

N-gram exploration

As mentioned by Hankare (2024), N-grams consist of consecutive 'n' words that capture contextual relationships in text. By preserving word order and context that single-word analysis loses, they are essential components in NLP tasks, especially for language modeling, predictive text, machine translation, and sentiment analysis (Hankare, 2024).

They are effective tools for extracting features in text classification tasks due to their ability to capture meaningful phrases and local linguistic patterns. The purpose of these is to assess quality by evaluating data preprocessing and identifying domain-specific language patterns across different datasets.

The complete set of 3-gram analysis can be found in the Appendix A.3.

By using a 3-gram analysis, it was possible to identify complex reasoning patterns in film criticism. In here, critics frequently use conditional phrases like 'could have been' and 'would have been' to explore hypothetical scenarios in films. In addition, film critics consistently use experiential markers like 'have ever seen' and 'have ever watched', which suggests that film reviews are fundamentally comparative.

By analyzing Amazon reviews closely, it is possible to observe how individuals discuss and recommend products. By demonstrating product experience, reviewers often use the phrase 'I have been' to build their credibility. Using strong evaluating language, like 'I highly recommend' and 'I absolutely love', is an explicit way to validate, while phrases like 'waste your money' are used as restrictions.

Fantasy or gaming-related terms dominate the vocabulary found in Twitter data. It includes phrases like 'found transponder snailgiants', 'transponder snailgiants sea', 'snailgiants sea monsters' and 'sea monsters other'. These terms are centered on fictional creatures or game elements. Social media engagement patterns such as 'click here watch' and 'here watch movie' reflect the platform's interactive and multimedia sharing culture.

Lastly, restaurant evaluation patterns appear in both the Yelp and SemEval datasets. The phrase "I have been" helps establish customer credibility by repeating experiences. By using the phrase 'I would have', Yelp reviews combine experiential storytelling with conditional reasoning. It also emphasizes social dining elements, shown by expressions like 'we were seated.', while 'I don't know' shows uncertainty. SemEval offers a superior collection with

specific hospitality terms like 'at reasonable prices' and 'food very good', indicating a structured approach to sentiment analysis, aimed at industry research rather than content created by consumers.

4.3 Data Preparation

Data Preparation is a crucial process to prepare the datasets for modeling, which requires the processes of cleaning, preprocessing, and feature extraction. As a result, this step demands a significant amount of effort as it has a significant influence on the model's overall performance (Data Science Process Alliance, 2024; Schröer et al., 2021).

As mentioned above, data preparation in sentiment analysis involves breaking text into tokens, removing unnecessary elements, standardizing text to lowercase, handling negations, and reducing words to their base form through stemming or lemmatization. Additionally, there is also the need to transform the text into numerical representations by utilizing approaches such as BoW, TF-IDF, or word and contextual embeddings.

To maintain a clearer structure, the preprocessing steps for each dataset will be detailed separately, followed by the specific preprocessing requirements for each deep learning model.

4.3.1 Dataset Specifications

The preparation of the T4SA dataset included merging two datasets: one containing only the tweet IDs ('TWID') and raw text, and the other containing probability scores for each sentiment class. Then, a function was created to label tweets based on their highest scores to categorize them into a sentiment class. The tweets undergo preprocessing by removing URLs, mentions, hashtags, punctuation, numbers, and stopwords, filtering out retweets, and ensuring that there are no sentiment labels that are absent. Finally, the sentiment labels were one-hot encoded to prepare the data for the multi-class classification task.

The data preprocessing in the Amazon dataset began with the removal of unnecessary columns that did not contribute to sentiment analysis, namely, images, parent_asin, user_id, asin, timestamp and helpful_vote. In this way, only the relevant information was kept to make the analysis more efficient. In addition, the ratings ranged from 1 to 5 and they were converted into three class labels: negative (1 and 2), neutral (3) and positive(4 and 5).

To match the sentiment classification schema used in other datasets, the Yelp dataset went through label normalization. In the beginning, Yelp reviews were rated on a scale of 0 to 4. The ratings were merged into three sentiment classes, which consisted of negative (0-1), neutral (2), and positive (3-4).

Four classes of sentiment labels were originally used in the SemEval 2014 task: conflict, negative, neutral, and positive. The conflict and the negative class were merged to ensure consistency with other datasets that use three sentiment categories. Additionally, the missing values of AspectTerm were filled with empty strings and missing values in Polarity were assigned a neutral label. In the end, concatenating sentences with their associated aspect terms, with the 'SEP' token separated, was done to prepare data for ABSA models. By processing both the full sentence and the specific aspect together, the model can understand the relationship between the specific aspect and the sentiment expressed towards it.

Lastly, the IMDB dataset consisted of only two classes: positive and negative. Thus, no further modifications were needed as it was sufficient for the sentiment class. It is important

to consider it during the following steps as it is the only task that requires binary classification.

Dataset Splitting

Initially, the dataset was divided into 80% for training, 20% for testing using stratified sampling to ensure balanced sentiment class distribution. Afterwards, 20% of the training set was made available for validation, resulting in a final ratio of 64% training, 16% validation, and 20% testing. In addition, in larger datasets a random selection of 100 000 samples from the training set was made for model training to manage computational resources effectively.

4.3.2 LSTM and CNN Preparation

Text Pre-processing

CNN and LSTM employed the Keras Tokenizer to convert text into integer sequences, while restricting it to the top 10 000 most frequently used words in the vocabulary. To make sure all sequences were 100 tokens long, the padding process was performed, and then the sentiment labels were converted to one-hot encoded vectors.

Word Embeddings

These models used GloVe embeddings to convert words into numerical vectors. In this way, an embedding matrix that links every word in the vocabulary to the pre-trained vector was created. The weights for the embedding layer will be determined by this.

As mentioned in the literature review, traditional sentiment analysis approaches include lowecassing, stopword removal, stemming and others. However, since GloVe was trained on raw text that preserved punctuation and stopwords it can handle text effectively with minimal pre-processing. Additionally, deep learning models are effective when learning important features without the need for stemming or stopwords removal.

4.3.3 BERT Preparation

The modelling approach for BERT requires a specific format of data to be applied to the different datasets.

Tokenization

In order to convert the text to BERTs input format, the pre-trained BERT tokenizer ”distilbert-base-uncased” was utilized. This tokenizer is responsible for converting words to token, helping the model dealing with variations in word morphology and out-of-vocabulary words. As explained by Devlin, Chang, Lee, and Toutanova (2019), special tokens such as [CLS] and [SEP] are added to help the model understand the structure of the data . The first one is added at the beginning of the sentence in classification tasks, while [SEP] is added in the end to separate pairs of sentences (Devlin et al., 2019).

Following that, to standardize the fixed inputs needed for training deep learning models, truncation and padding were done. Truncation reduces long sequences to the pre-determined maximum length (set to 256), while padding extends short sequences with padding tokens to reach the defined length.

In the end, the tokenizer automatically generates attention masks, preventing the model for using padding information in the calculations, and then these tokenized inputs are transformed into dictionaries.

Dataset Integration

Here, the tokenized text data is transformed into an optimized data stream for training BERT. This was achieved by creating TensorFlow datasets that pair tokenized inputs (IDs and attention masks) with their corresponding sentiment labels. In order to manage memory efficiently, the data was processed into complete datasets by splitting it up into 10 000 samples. Additionally, the training was optimized using the following techniques:

- the datasets were shuffled to randomize the order of examples, with the intention of preventing sequence bias;
- Combining data into batches (set to 32) to perform parallel processing;
- Prefetching was applied to load the next batch while the current batch is being processed, to improve efficiency (TensorFlow, n.d.);

This approach allows feeding properly formatted inputs to the BERT model at the same time that maximizes computational efficiency. However, it's important to note that this is used in validation and test datasets but without the shuffling part, since it doesn't affect the results in the evaluation phase.

4.4 Modeling

In the modeling phase, the selection of appropriate algorithms is made based on the type of problem. The data is sorted into training and validation sets, and the actual models are created. The final step is to gain insight into their performance.(Data Science Process Alliance, 2024; Schröer et al., 2021)

Therefore, once the final datasets have been prepared, the LSTM and CNN models will be trained from scratch. Subsequently, pre-trained BERT models will be utilized and fine-tuned to align with the domain and characteristics of the datasets.

Once all preprocessing steps are complete, it's time to proceed to the model training phase. To create a solid foundation for comparative analysis, similar training methodologies were implemented across all datasets in the different models.

4.4.1 CNN

To perform multi-class sentiment classification, the CNN were implemented using the Keras functional API within Tensorflow. In this way, it was possible to extract the features from text sequences by leveraging pre-trained word embeddings and multiple convolutional operations.

For text classification, CNNs typically have an embedding layer, convolutional and pooling operations, and fully connected layers (GeeksforGeeks, 2025b). In this study, the CNN's structure was as follows:

1. **Embedding layer (with pre-trained Glove embeddings)**

The model begins with an input that accepts text sequences of a fixed length. These sequences were processed through an embedding layer utilizing pre-trained GloVe word vectors. By incorporating this pre-trained embeddings, the model benefits from semantic knowledge about

words based on their contextual relationships in real world text. Here, the embedding weights were frozen during training to preserve their learned representations.

2. **Convolutional and Pooling Layers**

Following the embedding layer, the text representation goes through a series of three 1D convolutional layers. Each one of them contains 128 filters with a kernel size of 3 and using ReLU activation to introduce non-linearity. After the first two convolutional layers, MaxPooling layers with a pool size of 3 are applied to downsample the feature maps and only retain the main information. After that, a Global Max Pooling operation condenses the entire feature map into a single vector by extracting the maximum value from each feature map. In this way, it is possible to ensure that the important features are still preserved while reducing dimensionality.

3. **Fully Connected and Output Layers**

The algorithm ends with a Dense output layer using softmax activation, which enables the production of probability distribution across all sentiment classes. Here, the number of units corresponds to the number of sentiment categories being classified. Finally, the output layer produces the probability distribution over the classes, allowing the sentiment classification.

4.4.2 LSTM

The architecture implemented for LSTM consisted of the following key components:

1. **Embedding layer (with pre-trained Glove embeddings)**

In order to capture semantic relationships between words, LSTM uses an embedding layer that transforms word indices into continuous vector spaces. The vector space is closely grouped with words with similar meanings in the embedding layer. The model's ability to capture relationships between words is determined by the size of these vectors

2. **LSTM layer**

Dual LSTM layers were used to learn long-term dependencies in sequential data. A hierarchical feature extraction process is established by the first layer, which returns the entire sequences, processes the embedded text, and sends the entire sequence of hidden states to the second layer. Following that, the second layer focuses on condensing the sequential information into a fixed size representation.

3. **Dense Layer**

Following that, a dense layer with ReLU activation fed these representations, which enabled the algorithm to learn complex patterns from the LSTM output.

4. **Output Layer**

The final models differ depending on whether it is a binary or multiclass classification task. In the output layer, the model adopts 3 neurons with softmax activation for the multiclass classification task, whereas for binary classification typically uses a single neuron with sigmoid activation. Finally, the last dense layer, where softmax activation is employed as a classification method, converts the extracted features into prediction probabilities.

4.4.3 BERT

As mentioned in the previous chapter, during BERT modeling, the 'distilbert-base-uncased' NLP model was used and tuned to perform text classification tasks. Despite the fact that it is a distilled version of BERT, the language comprehension capabilities are almost intact in this version, retainning approximately 97% of the language comprehension abilities (Sanh, Debut, Chaumond, & Wolf, 2020).

In addition, Sanh et al. (2020) found that the DistilBert is 40% smaller and operates 60% faster. As a result, it was the most practical choice for this project because of the limited computational resources. Furthermore, the "uncased" version ensures that the same words but with different capitalization are treated in the same way.

The architecture implemented consists of the following key components:

1. **Base Model**

As mentioned before, DistilBERT takes the reviews text and transforms it into vectors, splitting the input text into tokens that are then passed through the transformer layers. These tokens pass through multiple attention mechanisms to allow capturing contextual relationships, regardless of their distance in text (since it understands long-range dependencies). This allows the network to understand certain nuances like negation and/or domain specific expressions.

2. **Feature Pooling**

GlobalAveragePooling1D was crucial for turning the word level meanings into a sentence level meaning, since it averages all token vectors in a sentence to create a unique vector that summarizes the meaning of the full sentence (Stack Overflow, 2023). In this way, it gives a semantic summary and it becomes more efficient since the number of parameters is reduced.

3. **Regularization**

In order to prevent overfitting, dropout layers with a rate equal to 30% were implemented. By randomly deactivating 30% of the neurons during training, these layers prevent the model from relying too heavily on certain neurons (Kashyap, 2024). In this way, it encourages the model to be more robust by forcing it to rely on multiple directions. Two dropout layers were placed after the pooling and after the hidden layer. The first one helps to prevent the model of relying on certain aspects of the sentence representation, while the second one prevents the hidden layer from becoming too dependent on a small subset of neurons.

4. **Dense Layer**

To connect each input to every output, a dense layer with 256 neurons and ReLU activation was employed. This layer helps to translate the general understanding of text from the pre-trained model into a sentiment aware format, while handling complexity when the meaning is very subtle.

5. **Output Layer**

Similarly to the architecture of CNN and LSTM, the output layer uses a softmax function to transform the raw output of the model into probabilities, with the class with the highest probability being the predicted sentiment. As a result, the model's level of confidence in its prediction can be determined.

4.4.4 Models Setup

Both LSTM and CNN architectures were configured with identical parameters during the model setup phase to guarantee fair comparison. The text was preprocessed by tokenizing with a maximum vocabulary size of 10 000 words, and padding or truncating sequences to a uniform length of 100 tokens. Additionally, the semantic relationships between words were captured by using 100 dimensional word embeddings in both models. To monitor performance,a validation split of 20% when training with 128 samples and 10 epochs was implemented.

1. **Loss function**

Loss functions are mathematical expressions that are used to predict how well the deep learning models are performing by measuring the difference between the predictions and actual labels (Bergmann & Stryker, 2024). Therefore, for each model the loss function was selected based on the characteristics of each dataset. For the CNN model, the categorical cross entropy was used since in the preprocessing one-hot encoded labels were applied. For BERT models applied to multi-class classification, where the output layers employed the function softmax with three neurons and the labels were integer encoded, the chosen loss function was sparse categorical cross entropy. In the case of binary classification (in the IMDB dataset), the model employed a single output neuron with sigmoid activation, and thus the binary cross entropy was applied.

2. **Optimizer**

An optimizer is also essential in deep learning models to adjust the models weights to minimize the loss function, measuring how far are the predictions from the actual results (Ultralytics, 2025).

In the modelling phase,the optimizer chosen for all models was the Adam (Adaptive Moment Estimation) optimizer due to the capability to adapt the learning rate dynamically during training (Ultralytics, 2025). However, when working on fine tuning large pre-trained models like BERT it is recommended to use a smaller learning rate (3e-5) to ensure stable updates to the pre-trained weights.

3. **Epochs and Batch size**

The total number of iterations of all training data passed through the machine learning model is referred to as an epoch (GeeksforGeeks, 2025a). However, processing the entire dataset can be inefficient and/or expensive in terms of memory limitations, being a common approach to divide the dataset into small batches (Devansh, 2022).

In this way, the number of epochs was initially set to 10 to allow the model enough time to learn from the data. Due to the high computational cost and GPU memory requirements of the BERT models, this value was decreased to 5, whereas for the SemEval dataset it was set to 20 since the dataset is relatively small.

Due to these requirements different batch sizes were also used based on the characteristics of the different models. Normally, models like LSTM and CNN, which were trained from scratch, tend to operate better with smaller batch sizes. On the contrary, BERT models are capable of handling larger batches efficiently allowing us to have a better gradient estimate and reduce noise. In this way, a batch size equal to 32 was used in LSTM and CNN, while a larger batch size of 64 was applied in BERT models.

4. **Callbacks**

Callbacks were used to make the training process more automated and optimize the training process across all models (Keras, 2025). The following strategies were applied:

- **Model Checkpoint**, allowing to save the models best weights during training, based on the highest validation accuracy (Keras, 2025).
- **Early Stopping**, which is a technique that enables stopping the training process when the model's performance on a certain metric stops improving for a determined number of consecutive epochs (Keras, 2025). In this case the metric used was the validation accuracy and training was stopped after three consecutive epochs without improvements, helping speed up training while reducing the risk of overfitting at the same time.

After the training is completed, the weights of the best models were saved for testing and evaluation. For BERT models, it was important to save the entire model (including architecture and weights) to ensure that the fine-tuned transformer setup was restored properly.

4.5 Evaluation

To determine if the success criteria defined in the Business Understanding phase were met, the model performance is assessed in the fifth phase of the CRISP-DM framework. The evaluation phase involves assessing whether the model meets the objectives and goals defined in the initial phase of the project. Furthermore, a comprehensive examination of the entire process is carried out to guarantee that there are no corrections required. The decision on whether to move to the next phase, continue iterating on the models, or change the approaches is made here (Schröer et al., 2021).

Thus, this analysis will consist of comparing key classification metrics and performance efficiency. Accuracy, precision, recall, F1-score, macro-F1, and weighted-F1 are the primary metrics taken into account. As mentioned before, these metrics are described as follows:

- **Accuracy** contains information about the overall classification performance and is commonly reported in sentiment analysis literature as a fundamental metric. Although useful for assessing general performance, accuracy can be deceptive when dealing with class-imbalanced datasets, as it may overestimate the performance of majority classes (Rathi, 2024).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.1}$$

- **Precision** measures the accuracy of positive sentiment predictions, which is the percentage of true positive predictions among all positive predictions:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.2}$$

- **Recall or Sensitivity** is a measure of the model's capacity to identify all true positive sentiments:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

- **F1-score** reflects the balance between precision and recall, being useful when the class distribution is imbalanced. For multi-class data, the F1-score is based on calculating individual class scores and averaging them to generate an overall score (Kundu, 2022).

$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

- **Macro-F1** treats all sentiment classes (negative, neutral, positive) equally, regardless of their frequency in the dataset. This prevents artificially high scores from being obtained by simply favoring majority classes (Kumar, 2023; Kundu, 2022).

$$\text{Macro-F1} = \frac{1}{n} \sum_{i=1}^{n} \text{F1}_i \tag{4.5}$$

- **Weighted-F1** compensates for the actual class distribution in the dataset, averaging the class-wise F1-scores according to their support. This provides a performance measure that more accurately reflects real-world scenarios where class imbalance naturally occurs (Kundu, 2022).

$$\text{Weighted-F1} = \sum_{i=1}^{n} w_i \times \text{F1}_i \tag{4.6}$$

  In this case, w_i corresponds to the proportion of samples in class i relative to the total number of samples (Kumar, 2023).

In here, the evaluation of the model's testing accuracy, F1 score, confusion matrix, and AUC score is done. Firstly, it measures accuracy and inference time, followed by generating class predictions. Then, it creates a classification report, F1 scores (for each class, macro, and weighted), and the confusion matrix. Finally, ROC-AUC is computed for every class and the mean AUC, providing a complete picture of the model's performance.

In summary, when these metrics are combined, they establish a comprehensive evaluation framework that balances performance assessment without regard to class and actionable insights for sentiment analysis applications.

Chapter 5

Experiments and Results

The objective of this study is to draw conclusions on two research questions that influence the current state of sentiment analysis research. These research questions are formulated as follows:

- What is the potential of deep learning architectures such as LSTM, CNN and BERT in sentiment analysis? How do they compare in terms of performance?
- How can we adapt BERT to aspect-based sentiment analysis? What steps can be taken to improve the detention of sarcasm, negations and multiple-aspect sentiments?

5.1 Potential of deep learning architectures in sentiment analysis

RQ1: What is the potential of deep learning architectures such as LSTM, CNN and BERT in sentiment analysis? How do they compare in terms of performance?

The primary research question is designed to evaluate the strengths and limitations of various deep learning approaches for sentiment analysis. To comprehend the best performance of each architecture under different conditions, the objective is to develop a comparative baseline using multiple evaluation metrics. The LSTM aptitude for processing things sequentially, CNN aptitude for deriving features through convolution operations, and BERT aptitude for comprehending context in both directions are all included. By doing this, it is possible to compare their relative effectiveness across sentiment analysis challenges.

5.1.1 Performance Comparison

Table 5.1: Comparison of Model Performance Across Datasets

Dataset	Model	Accuracy	Precision*	Recall*	F1-score*	Macro-F1	Weighted-F1	Train/Inference Time
IMDB	CNN	0.8406	0.83/0.85	0.86/0.82	0.84/0.84	0.8406	0.8406	**147.81s / 6.29s**
	LSTM	0.8499	0.88/0.83	0.82/0.88	0.84/0.85	0.8497	0.8497	392.83s / 34.24s
	BERT	**0.9074**	**0.92/0.90**	**0.89/0.92**	**0.91/0.91**	**0.9074**	**0.9074**	2278.84s / 211.83s
Amazon	CNN	0.8561	0.76/0.46/0.89	0.77/0.13/0.96	0.76/0.20/0.93	0.6299	0.8351	**2165.89s / 40.96s**
	LSTM	0.8828	0.79/**0.48**/0.93	0.86/0.24/0.96	0.82/0.32/**0.95**	0.6962	0.8711	17382.67s / 92.59s
	BERT	**0.8896**	**0.81/0.48/0.94**	**0.87/0.26/0.97**	**0.84/0.33/0.95**	**0.7080**	**0.8789**	14495.79s / 1282.09s
Yelp	CNN	0.7713	0.79/0.55/0.83	0.89/0.39/0.84	0.83/0.45/0.84	0.7088	0.7596	**5097.07s / 41.57s**
	LSTM	0.7982	0.82/0.61/0.84	**0.90**/0.43/0.88	0.86/0.51/0.86	0.7413	0.7882	13219.55s / 87.63s
	BERT	**0.8372**	**0.88/0.63/0.89**	0.89/**0.61/0.89**	**0.89/0.62/0.89**	**0.8006**	**0.8366**	50903.25s / 1231s
T4SA	CNN	0.9438	0.90/0.95/0.96	0.85/0.97/0.96	0.87/0.96/0.96	0.9301	0.9435	**5855.62s / 51.42s**
	LSTM	0.9500	**0.92**/0.95/0.96	0.85/0.97/**0.97**	0.88/0.96/**0.97**	0.9370	0.9495	41514.05s / 264.83s
	BERT	**0.9620**	**0.92/0.97/0.98**	**0.92/0.98**/0.96	**0.92/0.97/0.97**	**0.9544**	**0.9620**	34044.87s / 765.89s
SemEval	CNN	0.7500	0.71/0.79/0.73	0.48/0.71/0.90	0.57/0.75/0.81	0.7087	0.7416	**50.59s / 0.80s**
	LSTM	0.8040	0.64/0.87/0.84	0.72/0.80/0.84	0.68/0.83/0.84	0.7824	0.8060	338.70s / 2.61s
	BERT	**0.8602**	**0.75/0.93/0.85**	**0.73/0.82/0.95**	**0.74/0.87/0.90**	**0.8365**	**0.8590**	295.16s / 4.09s

.

*These columns show results divided into negative / neutral / positive classes, except for the IMDB dataset, where results are divided into negative / positive only (binary classification)

IMDB

The model that excelled at sentiment classification using the IMDB dataset was BERT. It achieved a 90.74% accuracy, but required the most time to process. On the contrary, CNN achieved an accuracy of 84.06% (lower than BERT but still a good performance), while offering the fastest training and inference time, suggesting that it could be useful in situations where resources are limited. With longer training time, LSTM demonstrated a modest improvement over CNN.

To sum up, all models proved to have a good balance between negative and positive classes in precision and recall metrics. The data revealed that there was a significant increase in computational demands, with BERT requiring approximately 15 times longer training time compared to CNN.

Amazon

On the Amazon dataset, BERT was the most accurate (88.96%), slightly ahead of LSTM (88.28%) and notably ahead of CNN (85.61%). In precision, recall, and F1-scores, it performed better, especially for the positive class, while BERT and LSTM were superior to CNN in the negative class. The neutral class was a challenge for all models, with low precision (around 0.46–0.48) and recall (below 0.30), which reflects the challenge of capturing neutral sentiment.

Despite BERT's superior performance, it came with the heaviest computational cost, requiring 14 495 seconds for training and 1282 seconds for inference, compared to LSTM's slightly longer training time but much faster inference, and CNN's overall speed advantage (2165 seconds training, 40 seconds inference). The trade-off is that BERT provides the best performance, but CNN and LSTM remain attractive for faster, resource efficient tasks where slight accuracy reductions are acceptable.

Yelp

BERT once more achieved the highest accuracy (83.72%) on this dataset, surpassing LSTM (79.82%) and CNN (77.13%). Precision, recall, and F1-scores were areas where BERT excelled, particularly in the positive and negative classes, with the gap between models narrowing in the neutral class. BERT's balanced performance across classes was confirmed by their higher macro-F1 (80.06%) and weighted-F1 (83.66%).

However, this strong performance came with significant computational demands. BERT required over 50 900 seconds for training and 1231 seconds for inference, far exceeding LSTM 13 219 seconds and CNN 5 097 seconds for training. The fastest for inference was still CNN, with a time of 41.57 seconds.

T4SA

The T4SA dataset was the dataset on which all models performed best. The highest overall accuracy (96.20%) was achieved by BERT, just like the other cases. Despite this, all models were able to predict the different sentiment classes with strong performance. However, BERT accomplishes this with a significant increase in computational cost. LSTM, on the other hand, has a strong balance between accuracy and efficiency, which makes it a more viable option in resource constrained scenarios.

SemEval

The CNN model achieved a 75% accuracy rate with a faster training and inference time, which was accompanied by reasonable precision across all classes. The F1-score showed an imbalance in the dataset, with a lower recall for the negative class (0.48) but a strong recall for the positive class (0.90). When compared to CNN, LSTM demonstrated an improvement in

accuracy and more balanced performance across all classes. The F1-score is more consistent and the precision and recall are better. As seen with other datasets, BERT is the best performing model, achieving an accuracy of 86.02%, strong precision across all classes, and the best F1 scores overall. The high recall, especially for the positive class (0.95), was also shown despite the fast training and slow inference.

5.1.2 Auto-rank framework

According to Herbold (2020), Autorank is a statistical framework that is designed to efficiently compare multiple paired populations, through statistical testing. In this study, Autorank is going to be used to compare multiple deep learning models (CNN, LSTM and BERT) across multiple datasets.

1. **Accuracy Performance Comparison**

At a significance level of 5%, a statistical comparison of the CNN, LSTM and BERT models was performed across the five different datasets. The following statistic tests were performed:

- Shapiro–Wilk normality test: At a significance level of 5%, the null hypothesis was not rejected (p-value=0.529), meaning that all populations can be considered normally distributed.
- Bartlett's test for homogeneity of variances: At a significance level of 5%, the null hypothesis was not rejected (p-value=0.529), confirming the homoscedastic of the data.

Since more than two populations are being tested and normality and homoscedastic were confirmed by the Shapiro–Wilk and Bartlett's test, respectively. The ANOVA test was employed to see if there were significant differences between the mean performance of the models.

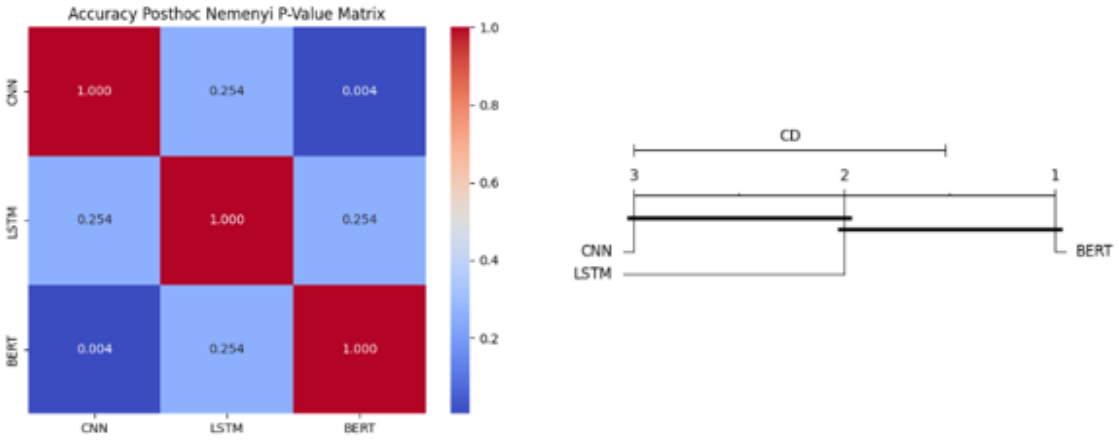

Figure 5.1: Accuracy: Post-hoc Nemenyi Test P-Value Matrix and Critical Difference Diagram

With a confidence interval range of around 81% to 98%, BERT exhibits the highest accuracy performance. This suggests that BERT's true mean accuracy on similar sentiment analysis tasks falls within this range, with 95% confidence. LSTM's confidence interval ranges from 75% to 95%, while CNN's accuracy ranges from 70% to 97%.

In this way, it was possible to observe the existence of statistical differences between some models. Therefore, the Post-hoc Nemenyi P-Value Matrix (Figure 5.2.) was employed to explore which pair showed significant differences, since it enables comprehensive statistical evidence for comparing the accuracy performances of the three models in pairwise comparisons.

Therefore, at 5% significance level, it was possible to confirm that BERT significantly outperforms CNN with a p-value of 0.0045, indicating strong statistical evidence that BERT's superior accuracy is not due to random chance.

Furthermore, it was also confirmed that there is no statistically significant difference between CNN and LSTM, and LSTM and BERT. Although BERT has a slightly lower mean rank, it is still statistically equivalent to LSTM with a p-value of 0.25.

In summary, these findings confirm that BERT demonstrates statistically superior performance compared to CNN, LSTM is statistically equivalent to CNN and LSTM and CNN shows the lowest performance.

2. **F1-score**

The same approach used for comparing accuracy performance was employed here, with the only difference that F1-score was analysed separately for positive and negative classes. Therefore, at the same significance level (5%), the following statistic test were performed:

- Shapiro–Wilk normality test: The null hypothesis was not rejected (Negative: p-value=0.115; Positive:p-value=0,096), meaning that all populations can be considered normally distributed.
- Bartlett's test for homogeneity of variances: The null hypothesis was not rejected (Negative: p-value=0.471; Positive: p-value=0.579), confirming the homoscedastic of the data.
- ANOVA test, in which the null hypothesis was rejected (Negative: p-value=0.011; Positive: p-value=0.004) indicating statistically significant differences between the model means.

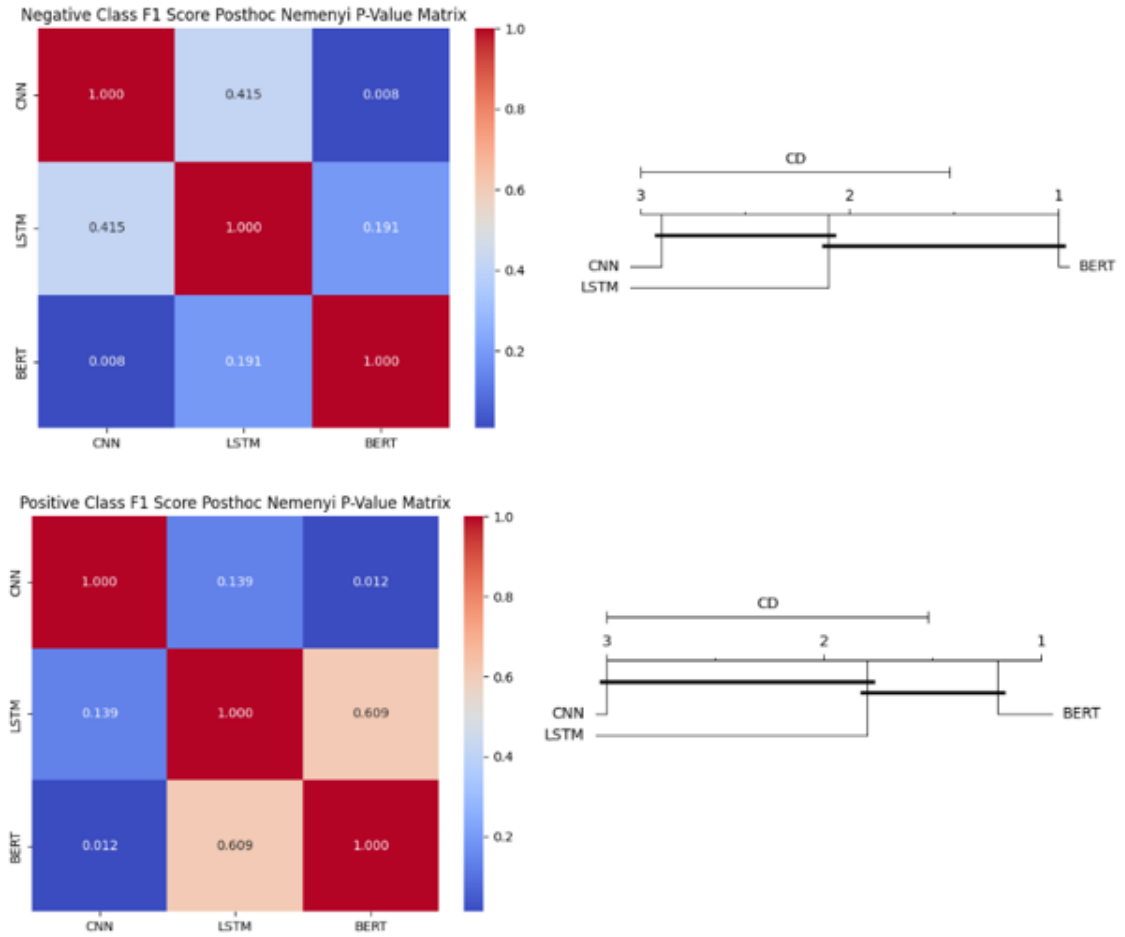

Figure 5.2: F1 score (negative and positive classes): Post-hoc Nemenyi Test P-Value Matrix and Critical Difference Diagram

At both negative and positive classes, statistically significant differences between CNN and BERT (Negative: p-value= 0,0075; Positive: p-value= 0,0012) were found, but no significant difference between the other model pairs. Therefore, BERT's stronger class predictions cannot be attributed to random chance.

3. **Macro-F1 and Weighted-F1**

At a significance level of 5%, the macro F1-score and weighted F1-score were evaluated using the same methodology. As presented in Appendix, Figure A.12 and Figure A.13, in both cases, the Shapiro-Wilk test confirmed that all populations are distributed in a normal way, and Bartlett's test confirmed that the data were homoscedastic. Additionally, ANOVA was conducted in both metrics analysis and revealed statistically significant differences between the model means (Macro-F1: $p = 0.004$; Weighted-F1: $p = 0.003$).

The posthoc analysis showed that in both cases BERT significantly outperformed CNN ($p = 0.0045$), but no significant differences were found between CNN and LSTM or between LSTM and BERT ($p = 0.25$ for both).

4. **ROC AUC**

Regarding the Area Under the Curve (ROC), at a significance level of 5%, the following statistics tests were performed across the different datasets:

- Shapiro–Wilk normality test: The null hypothesis was not rejected (p-value = 0.082), meaning that all populations can be considered normally distributed.
- Bartlett's test for homogeneity of variances: At a significance level of 5%, the null hypothesis was not rejected (p-value = 0.440), confirming the homoscedasticity of the data.
- Friedman test: The null hypothesis was rejected at a significance level of 5% (p=0.007), indicating that there is a statistically significant difference in central tendency between the models.

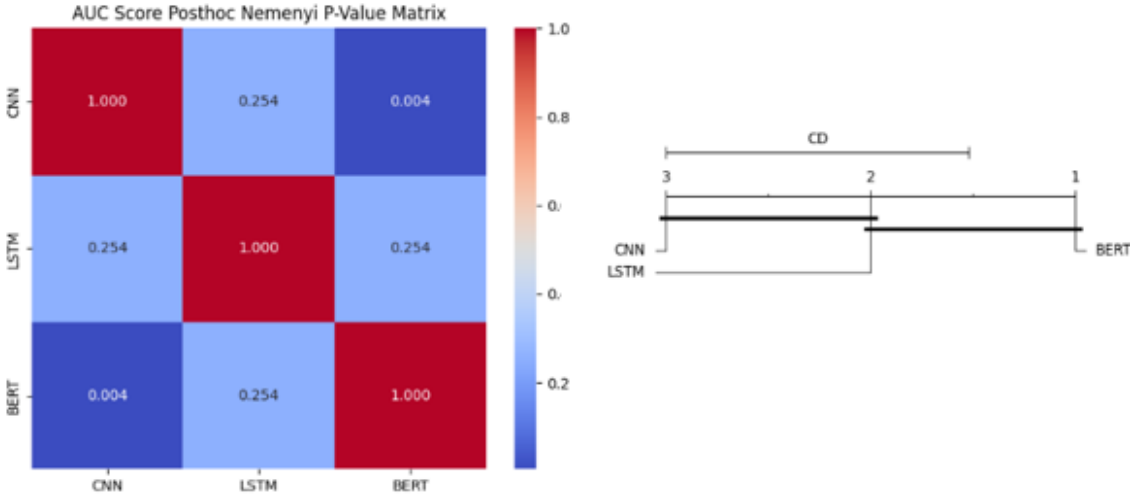

Figure 5.3: ROC AUC: Post-hoc Nemenyi Test P-Value Matrix and Critical Difference Diagram

In terms of AUC score, BERT emerged as the top-performing model, with a statistically significant advantage over CNN, but not LSTM. As shown in the Posthoc Nemenyi P-value Matrix, there is also a significant statistical difference between CNN and BERT (p=0.004), while the differences between the other pairs of models are not statistically significant. In a nutshell, the Critical Distance Diagram shows that only CNN and BERT exceed the critical distance for all metrics, verifying a statistically significant difference between them.

5.1.3 Confusion Matrix Performance

Table 5.2: Confusion Matrices for IMDB Dataset

CNN-IMDB	**Predicted Negative**	**Predicted Positive**
Actual Negative	85.66% (10708)	14.34% (1792)
Actual Positive	17.54% (2193)	82.46% (10307)
LSTM-IMDB	**Predicted Negative**	**Predicted Positive**
Actual Negative	81.63% (10204)	18.37% (2296)
Actual Positive	11.66% (1457)	88.34% (11043)
BERT-IMDB	**Predicted Negative**	**Predicted Positive**
Actual Negative	89.21% (11151)	10.79% (1349)
Actual Positive	7.72% (965)	92.28% (11535)

Regarding the sentiment classification task performed on the IMDB dataset, it can be observed that CNN shows an imbalance in error distribution, higher false negative rate (17.54%) than false positive rate (14.34%) In this way, the CNN algorithm shows a bias towards predicting negative classes, tending to underpredict positive classes. On the other hand, the LSTM model exhibits a more balanced performance, however it slightly leaned towards the positive classes, since the false positive rate (18.37%) is higher than the false negative rate (11.66%).

On the contrary, BERT shows the highest accuracy for positive sentiment detection (92.28%) and the lowest false negative rate (7.72%), suggesting that BERT is more balanced than the others (does not underpredict or over-predict positive sentiments).

Table 5.3: Confusion Matrices for Amazon Dataset

CNN-Amazon	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	77.18 % (22400)	2.95 % (857)	19.87 % (5766)
Actual Neutral	37.83çºl % (4261)	12.59 % (1418)	49.56 ° ° ° % (5582)
Actual Positive	2.92 % (2922)	0.80 % (804)	96.27 % (96296)
LSTM-Amazon	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	86.05% (12487)	5.27% (765)	8.68% (1260)
Actual Neutral	36.71% (2067)	23.69% (1334)	39.59% (2229)
Actual Positive	2.44% (1222)	1.36% (682)	96.19% (48107)
BERT-Amazon	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	86.72% (25170)	5.22% (1514)	8.06% (2339)
Actual Neutral	37.63% (4237)	25.58% (2881)	36.79% (4143)
Actual Positive	1.66% (1661)	1.59% (1589)	96.75% (96772)

On the Amazon dataset, CNN shows a clear bias towards positive sentiments, with only 77.18% accuracy for negative sentiment and 12.59% for neutral sentiment, while achieving 96.27% for positive sentiment,overestimating positives and negatively impacting the other sentiment classes. Additionally, the LSTM model offers a more balanced prediction than CNN, even though it still has an inclination towards the positive class.

BERT gives the most balanced performance across all three sentiment categories. This architecture achieves the highest accuracy rates for all three classes while maintaining better balance, indicating that BERT is a more robust and generalizable model for this dataset.

Table 5.4: Confusion Matrices for T4SA Dataset

CNN-T4sa	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	85.25 % (33934)	11.89 % (4731)	2.86 % (1138)
Actual Neutral	2.23 % (2720)	96.56 % (117729)	1.21 % (1471)
Actual Positive	1.63 % (1212)	2.67 % (1984)	95.70 % (71073)
LSTM-T4sa	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	84.91 % (16899)	11.73 % (2334)	3.36 % (669)
Actual Neutral	1.62 % (986)	97.19 % (59245)	1.20 % (729)
Actual Positive	1.13 % (419)	2.05 % (761)	96.82 % (35954)
BERT-T4sa	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	92.44 % (29692)	5.14 % (1650)	2.43 % (779)
Actual Neutral	1.90 % (1701)	97.52 % (87490)	0.59 % (525)
Actual Positive	1.52 % (864)	2.25 % (1283)	96.23 % (54854)

T4sa was the dataset that achieved the best performance across all metrics, in all different datasets. CNN's strong performance in neutral classification (96.56%) is offset by a noticeable bias towards positive predictions, which sometimes leads to misclassification of negative cases. The distribution of the LSTM model is more balanced across all three classes (84.91% negative, 97.19% neutral, and 96.82% positive), but it has lower absolute performance than BERT. BERT's performance is particularly impressive, achieving 92.44% accuracy for negative, 97.52% for neutral, and 96.23% for positive sentiments , which makes it the most balanced and robust model among all three.

Table 5.5: Confusion Matrices for Yelp Dataset

CNN-Yelp	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	89.10 % (17820)	6.53 % (1305)	4.38 % (875)
Actual Neutral	35.46 % (3546)	38.51 % (3851)	26.03 % (2603)
Actual Positive	6.60 % (1320)	8.94 % (1788)	84.46 % (16892)
LSTM-Yelp	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	89.87 % (23366)	6.47 % (1682)	3.66 % (952)
Actual Neutral	30.56 % (3973)	43.47 % (5651)	25.97 % (3376)
Actual Positive	4.46 % (1159)	7.60 % (1977)	87.94 % (22864)
BERT-Yelp	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	89.45 % (46514)	9.04 % (4701)	1.51 % (785)
Actual Neutral	20.02 % (5204)	61.36 % (15954)	18.62 % (4842)
Actual Positive	1.68 % (875)	9.15 % (4758)	89.17 % (46367)

The CNN model's performance on the Yelp sentiment classification task is decent, but it struggles to accurately classify neutral sentiments (38.51% accuracy), with a slight preference for negative predictions. This is improved by the LSTM model's balanced distribution across negative, neutral, and positive classes (89.87%, 43.47%, and 87.94% respectively), and higher correct predictions in all categories compared to CNN. The BERT model is superior to both CNN and LSTM, achieving 89.45% accuracy for negative, 61.36% for neutral, and 89.17% for positive sentiments and demonstrates superior balance and generalization. BERT's robustness make it the best model for sentiment classification on the Yelp dataset.

Table 5.6: Confusion Matrices for SemEval Dataset

CNN-SemEval	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	47.73 % (84)	18.18 % (32)	34.09 % (60)
Actual Neutral	6.43 % (22)	70.76 % (242)	22.81 % (78)
Actual Positive	2.82 % (12)	7.51 % (32)	89.67 % (382)
LSTM-SemEval	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	72.16 % (127)	9.09 % (16)	18.75 % (33)
Actual Neutral	8.77 % (30)	80.41 % (275)	10.82 % (37)
Actual Positive	10.09 % (43)	6.10 % (26)	83.80 % (357)
BERT-SemEval	**Predicted Negative**	**Predicted Neutral**	**Predicted Positive**
Actual Negative	72.63 % (130)	7.26 % (13)	20.11 % (36)
Actual Neutral	8.13 % (27)	81.63 % (271)	10.24 % (34)
Actual Positive	3.70 % (16)	1.39 % (6)	94.92 % (411)

The BERT model is the most effective model for SemEval sentiment classification task, excelling in positive sentiment detection (94.92% accuracy) and maintaining strong results in negative and neutral classifications (72.63% and 81.63% respectively). All three sentiment categories are evenly distributed by the LSTM model(72.16% negative, 80.41% neutral, and 83.80% positive), with neutral sentiment having the highest accuracy (when compared to

other models). The CNN model does not perform well on negative and neutral detection (47.73% and 70.76% respectively), but it achieves reasonable performance on positive sentiment (89.67% accuracy), though still underperforming compared to BERT and LSTM

5.2 BERT Adaptation for ABSA

RQ2: How can we adapt BERT to aspect-based sentiment analysis? What steps can be taken to improve the detention of sarcasm, negations and multiple-aspect sentiments?

The second research question is about ABSA's specific challenges, especially the complex language issues that prevent traditional sentiment analysis systems from functioning properly, including sarcasm and context-dependent expressions .The focus of this research question is on how BERT's transformer architecture can be modified to address these challenges. Significant improvements in understanding complex language phenomena may be achieved by making small changes to its construction and training.

5.3 Main Findings

RQ1: What is the potential of deep learning architectures such as LSTM, CNN and BERT in sentiment analysis? How do they compare in terms of performance?

- Key Statistical Findings

The statistical analysis, which included tests to check if the data is normally distributed and has the same variance, as well as other tests, shows a consistent pattern across all the ways it was evaluated it. BERT shows the greatest potential for sentiment analysis tasks, performing better than CNN in terms of accuracy, F1-score (for both positive and negative results), Macro-F1, Weighted-F1, and ROC AUC metrics. This is not just a coincidence, as the p-value results are always below 0.05 when comparing BERT to CNN,, which shows that there is a real link.

Additionally, according to the confusion matrix analysis, BERT is well balanced across various emotions, with minimal class bias and the highest true positive rates across negative, neutral, and positive emotions. BERT's ability to comprehend contextual relationships and dependencies in both directions in the text is highlighted here.

It is worth noting that there were no major differences between CNN and LSTM, or between LSTM and BERT, for most measures in the results. Despite BERT's higher performance in tests, the difference is not significant enough for statistical significance (p-values > 0.05), which means that LSTM and BERT perform about the same.

LSTM has a great deal of potential as a method of doing things that is fast and doesn't require as much computing power. It performs just as well as BERT ($p > 0.05$ across all measures). LSTMs are excellent at analyzing sentiment, and they balance their effectiveness with their processing requirements. While not statistically significant, LSTM outperforms CNN on average and provides more balanced predictions than CNN across various types of sentiment.

On the other hand, CNN's potential for sentiment analysis tasks is limited, as it consistently shows the lowest performance across all datasets and metrics. The statistical analysis

confirms that CNN's performance is inferior compared to both BERT and LSTM, with the statistically significant gap confirmed only for BERT and CNN, and with systematic classification biases particularly affecting neutral sentiment detection and showing tendencies to overpredict positive sentiments.

- Model-Specific Performance

BERT is the ideal model for all data sets due to its strength and balance. The confusion matrix analysis indicates that BERT is more effective at making balanced predictions across diverse types of texts, thereby eliminating the classification biases observed in other models. Therefore, BERT consistently gets the best outcomes for positive, negative, and neutral categories, while keeping misclassification errors to a minimum.

LSTM is a good choice as it performs statistically equally well against both CNN and BERT, making it a balanced alternative. The model's error distribution is more balanced than CNN, but it occasionally exhibits slight biases towards certain types of sentiment, depending on the dataset.

CNN has the worst performance among all options when it comes to measuring and analyzing data sets. According to the confusion matrix analysis, CNN's predictions have systematic biases. It tends to overpredict certain types of sentiment (especially positive sentiments) and has difficulty categorizing neutral sentiments.

- Dataset specific insights

The performance of the various datasets varied, with T4sa achieving the most effective results across all models and metrics. According to this, the performance of model models is greatly impacted by dataset characteristics, such as text length, vocabulary diversity, and sentiment distribution. It was a challenge to detect neutral sentiment across multiple datasets.

RQ2: How can we adapt BERT to aspect-based sentiment analysis? What steps can be taken to improve the detention of sarcasm, negations and multiple-aspect sentiments?

In order to use BERT for ABSA, minor but important preprocessing modifications were made: empty strings were used to replace missing values in AspectTerm; missing polarity values were mapped to the neutral class to preserve dataset integrity; and each sentence was combined with its respective aspect term, separated by the 'SEP' token, to guarantee that the model receives both the global sentence context and the localized aspect information. By making this adjustment, BERT is able to better capture the relationship between an aspect and the sentiment expressed towards it.

These adaptations display how BERT can be transformed into an ABSA capable model with a simple modification to the input representation. Although the transformer architecture is naturally adept at capturing contextual dependencies, aligning sentence-level context with explicitly aspect terms enhances its capacity to handle the fine-grained nature of ABSA tasks.

The experiments demonstrate BERT's superior performance in general sentiment classification across multiple datasets, with additional evaluation on the SemEval 2014 dataset specifically designed for ABSA, confirming its adaptability to more complex sentiment tasks.

- BERT's Advanced Architecture Benefits

BERT's exceptional ability to maintain balanced predictions across sentiment classes and minimize misclassification errors is demonstrated in the experimental results. BERT's balanced performance demonstrates its superior contextual understanding, which is essential for identifying subtle linguistics like sarcasm and negations. Understanding complex sentiment relationships within text is possible thanks to the bidirectional attention mechanism in BERT's architecture.

- Implications for more complex sentiment tasks

The superior performance of BERT in the SemEval dataset, especially its ability to handle neutral sentiments effectively (as demonstrated in confusion matrices), confirms that it serves as a baseline for ABSA. The model's ability to handle multiple-aspect sentiments where different aspects may carry different sentiment polarities is demonstrated by its robust performance across different datasets.

However, domain specific fine tuning with datasets containing sarcasm, negation, and multi-aspect examples is still necessary to enhance BERT's performance. These sophisticated adaptations can develop further the advanced contextual understanding capabilities of BERT, as evidenced by the study's findings. The model's balanced performance across different categories of sentiment establishes its potential for reliable detection of complex multi-aspect sentiments with appropriate adaptations and enriched training data.

Chapter 6

Conclusion

6.1 Final Remarks

This study demonstrated that deep learning models, specifically BERT, perform superior then other methods. Despite this, there are still some issues; it was difficult for the majority of the models to comprehend complicated emotions such as sarcasm, irony, and other confusing expressions.

The current sentiment analysis methods cannot provide a deeper understanding of context for complex language. Using models such as BERT in areas with limited resources is problematic due to their high computer power requirements. In this way, it is difficult to access and deploy these models in real-time applications due to the computational burden.

Furthermore, some limitations were observed in the datastes chosen, such as identifying neutral sentiments. It was a challenge for all models in the Amazon and Yelp datasets, as they were not very accurate at finding or correctly labeling them. Due to its smaller size and more complicated language, the SemEval dataset was particularly challenging, resulting in the performance of all models being worse. All models received the best results from the T4SA dataset, but CNN often predicted positive sentiments too often, sometimes mistakenly labeling negative comments as positive.

Lastly, this study was constrained by the fact that the SemEval was the only dataset that could be used to evaluate ABSA, while the other datasets only support general sentiment classification.

6.1.1 Future Research Directions

Future research should focus on a few important areas to improve sentiment analysis. The ability of models to handle sarcasm, negation, and complex emotions can be greatly improved by creating special training methods. The addition of more language features, emotional detection parts, or special attention systems that recognize small contextual hints might be a part of this.

By studying ways to make models smaller and more efficient, the problem of transformer-based models requiring too much computer power could be resolved. This would make advanced sentiment analysis available to more types of computing systems. By using techniques such as quantization, pruning, and knowledge distillation, models could be kept working effectively while consuming less computer power.

To demonstrate how emotions are expressed differently and how well models work across different areas, testing models on different types of data and languages from different cultures would be helpful. Lastly, investigating how to combine external knowledge and common sense reasoning could assist models in better comprehending hidden meanings in text and emotions that depend on context.

By pursuing these research directions, it could be develop sentiment analysis systems that are more reliable, efficient, and practically applicable, and can handle the full complexity of human language expression.

6.1.2 Overall Conclusion

BERT is the best option for sentiment analysis when the right technology is available, as demonstrated by this research, as it consistently performs well. When resources are limited, LSTM is the optimal option because it is faster and doesn't require as much computing power as BERT. Furthermore, CNN's poor performance and classification bias across different datasets and sentiment types make it a poor choice for sentiment analysis tasks.

The study concludes that BERT is suitable for more complex sentiment analysis tasks, such as analyzing aspects, detecting sarcasm, and handling negation. It is better at understanding context and categorizing sentiments more evenly across different datasets and sentiment types

Bibliography

Ahmed, M., Pan, S., Wen, B., Su, J., Zhang, W., & Liu, Y. (2022). Bert-asc: Implicit aspect representation learning through auxiliary-sentence construction for sentiment analysis. *ArXiv*. Retrieved from https://doi.org/10.48550/arxiv.2203.11702

Alhamid, M. (2020, December). *What is cross-validation?* https://towardsdatascience.com/what-is-cross-validation-60c01f9d9e75. (Towards Data Science - Medium, Accessed: 2-Feb-2025)

Al-Qablan, T., Mohd Noor, M. H., Al-Betar, M., & Khader, A. T. (2023, 08). A survey on sentiment analysis and its applications. *Neural Computing and Applications*, *35*, 1-35. doi: 10.1007/s00521-023-08941-y

Bergmann, D., & Stryker, C. (2024, July 12). *What is a loss function?* IBM Think (Artificial Intelligence section). Retrieved from https://www.ibm.com/think/topics/loss-function

Cahyadi, A., & Khodra, M. L. (2018). Aspect-based sentiment analysis using convolutional neural network and bidirectional long short-term memory. In *2018 5th international conference on advanced informatics: Concept theory and applications (icaicta)* (p. 124-129). doi: 10.1109/ICAICTA.2018.8541300

Chakkarwar, V., Tamane, S., & Thombre, A. (2023). A review on bert and its implementation in various nlp tasks. In *Proceedings of the international conference on applications of machine intelligence and data analytics (icamida 2022)* (p. 112-121). Atlantis Press. Retrieved from https://doi.org/10.2991/978-94-6463-136-4_12 doi: 10.2991/978-94-6463-136-4_12

Colliot, O. (Ed.). (2023). *Machine learning for brain disorders* (1st ed.). New York, NY: Humana. (Hardcover) doi: 10.1007/978-1-0716-3195-9

Das, S., Tariq, A., Santos, T., Kantareddy, S. S., & Banerjee, I. (2023). Recurrent neural networks (rnns): Architectures, training tricks, and introduction to influential research. In O. Colliot (Ed.), *Machine learning for brain disorders* (pp. 117–138). New York, NY: Springer US. Retrieved from https://doi.org/10.1007/978-1-0716-3195-9_4 doi: 10.1007/978-1-0716-3195-9_4

Data Science Process Alliance. (2024). *CRISP-DM*. https://www.datascience-pm.com/crisp-dm-2/. (Accessed: 3-Feb-2025)

Devansh. (2022, January 17). *How does batch size impact your model learning*. Geek Culture (Medium). Retrieved from https://medium.com/geekculture/how-does-batch-size-impact-your-model-learning-2dd34d9fb1fa

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding*. Retrieved from https://arxiv.org/abs/1810.04805

Diwan, T., & Tembhurne, J. (2022, 06). Sentiment analysis: a convolutional neural networks perspective. *Multimedia Tools and Applications*, *81*. doi: 10.1007/s11042-021-11759-2

Gama, J., Carvalho, A., Faceli, K., Lorena, A., & Oliveira, M. (2017). *Extração de conhecimento de dados*. Silabo.

GeeksforGeeks. (2025a, July 17). *Epoch in machine learning*. GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/machine-learning/epoch-in-machine-learning/

GeeksforGeeks. (2025b). *Text classification using cnn.* Retrieved from https://www.geeksforgeeks.org/nlp/text-classification-using-cnn/ (Accessed: 16 August 2025)

Hankare, O. (2024, March 8). *Exploring n-grams: the building blocks of natural language understanding*. Medium. Retrieved from https://ompramod.medium.com/exploring-n-grams-the-building-blocks-of-natural-language-understanding-d40a7a309d12

Herbold, S. (2020). Autorank: A python package for automated ranking of classifiers. *Journal of Open Source Software*, *5*(48), 2173. Retrieved from https://doi.org/10.21105/joss.02173 doi: 10.21105/joss.02173

Hochreiter, S., & Schmidhuber, J. (1997, 11). Long short-term memory. *Neural Computation*, *9*, 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Hua, Y., Denny, P., Wicker, J., & Taskova, K. (2024, 09). A systematic review of aspect-based sentiment analysis: domains, methods, and trends. *Artificial Intelligence Review*, *57*. doi: 10.1007/s10462-024-10906-z

IBM. (2024a). *Convolutional neural networks.* https://www.ibm.com/think/topics/convolutional-neural-networks. (Accessed: 18-Dec-2025)

IBM. (2024b). *Deep learning.* https://www.ibm.com/think/topics/deep-learning. (Accessed: 2-Dec-2025)

Islam, M., Kabir, M., Ab Ghani, N., Zamli, K., Zulkifli, N., Rahman, M. M., & Moni, M. (2024, 03). "challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach". *Artificial Intelligence Review*, *57*. doi: 10.1007/s10462-023-10651-9

Jin, Y., Cheng, K., Wang, X., & Cai, L. (2023, 07). A review of text sentiment analysis methods and applications. *Frontiers in Business, Economics and Management*, *10*, 58-64. doi: 10.54097/fbem.v10i1.10171

Karimi, A., Rossi, L., & Prati, A. (2020). Improving bert performance for aspect-based sentiment analysis. *arXiv preprint arXiv:2010.11731*.

Kashyap, P. (2024, October). *Understanding dropout in deep learning: A guide to reducing overfitting.* Retrieved from https://medium.com/@piyushkashyap045/understanding-dropout-in-deep-learning-a-guide-to-reducing-overfitting-26cbb68d5575 (Accessed 2025-08-16)

Keras. (2025). *Callbacks api.* Keras 3 API documentation. Retrieved from https://keras.io/api/callbacks/ (Acesso em 15 de agosto de 2025)

Kumar, S. (2023). *Sharpening your model's performance: A deep dive into macro-f1 and weighted-f1 scores in multi-class classification.* https://medium.com/@shivakumarg814/sharpening-your-models-performance-a-deep-dive-into-macro-f1-and-weighted-f1-scores-in-b98339eb9dbb. (Accessed: 2025-08-30)

Kundu, R. (2022, December 16). *F1 score in machine learning: Intro & calculation.* V7 Labs blog. Retrieved from https://www.v7labs.com/blog/f1-score-guide

Li, X., Bing, L., Zhang, W., & Lam, W. (2019). Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883*.

Mao, Y., Liu, Q., & Zhang, Y. (2024). Sentiment analysis methods, applications, and challenges: A systematic literature review. *J. King Saud Univ. Comput. Inf. Sci.*, *36*, 102048. Retrieved from https://api.semanticscholar.org/CorpusID:269540930

Marcacini, R., & Silva, E. (2021, July). Aspect-based sentiment analysis using BERT with

disentangled attention. In *Latinx in ai at international conference on machine learning 2021*. doi: 10.52591/lxai2021072416

Mulyo, B. M., & Widyantoro, D. H. (2018). Aspect-based sentiment analysis approach with cnn. *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 142-147. Retrieved from https://api.semanticscholar.org/CorpusID:195831681

Rathi, D. (2024, October 14). *Handling imbalanced data: Key techniques for better machine learning*. Medium. Retrieved from https://medium.com/@dakshrathi/handling-imbalanced-data-key-techniques-for-better-machine-learning-6e33b466f8b7

Ruder, S., Ghaffari, P., & Breslin, J. G. (2016a). A hierarchical model of reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02745*.

Ruder, S., Ghaffari, P., & Breslin, J. G. (2016b, June). INSIGHT-1 at SemEval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis. In S. Bethard, M. Carpuat, D. Cer, D. Jurgens, P. Nakov, & T. Zesch (Eds.), *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)* (pp. 330–336). San Diego, California: Association for Computational Linguistics. Retrieved from https://aclanthology.org/S16-1053/ doi: 10.18653/v1/S16-1053

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). *Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter*. Retrieved from https://arxiv.org/abs/1910.01108

Schmitt, M., Steinheber, S., Schreiber, K., & Roth, B. (2018). Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks. *arXiv preprint arXiv:1808.09238*.

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying crisp-dm process model. *Procedia Computer Science*, *181*, 526-534. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050921002416 (CENTERIS 2020 - International Conference on ENTERprise Information Systems / ProjMAN 2020 - International Conference on Project MANagement / HCist 2020 - International Conference on Health and Social Care Information Systems and Technologies 2020, CENTERIS/ProjMAN/HCist 2020) doi: https://doi.org/10.1016/j.procs.2021.01.199

Shewalkar, A. N., Nyavanandi, D., & Ludwig, S. A. (2019). Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru. *Journal of Artificial Intelligence and Soft Computing Research*, *9*, 235 - 245. Retrieved from https://api.semanticscholar.org/CorpusID:201718834

Singh, R. (2024, November 7). *Long short-term memory (LSTM) networks*. https://medium.com/@RobuRishabh/long-short-term-memory-lstm-networks-9f285efa377d. (Medium)

Stack Overflow. (2023, January). *What does globalaveragepooling1d do in keras?* Retrieved from https://stackoverflow.com/questions/75067335/what-does-globalaveragepooling1d-do-in-keras (Accessed 2025-08-16)

Su, Z. (2024, 10). Applications of bert in sentimental analysis. *Applied and Computational Engineering*, *92*, 147-152. doi: 10.54254/2755-2721/92/20241711

Tang, D., Qin, B., Feng, X., & Liu, T. (2015). Effective lstms for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*.

Tang, D., Qin, B., & Liu, T. (2016). Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*.

Tay, Y., Tuan, L. A., & Hui, S. C. (2018). Learning to attend via word-aspect associative

fusion for aspect-based sentiment analysis. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32).

TensorFlow. (n.d.). *Better performance with the tf.data api*. Retrieved from https://www.tensorflow.org/guide/data_performance (Accessed: 2025-08-16)

Ultralytics. (2025, July 1). *Adam optimizer*. Ultralytics Glossary. Retrieved from https://www.ultralytics.com/glossary/adam-optimizer

Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016, November). Attention-based LSTM for aspect-level sentiment classification. In J. Su, K. Duh, & X. Carreras (Eds.), *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 606–615). Austin, Texas: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D16-1058/ doi: 10.18653/v1/D16-1058

Wnkhade, M., Rao, A., & Kulkarni, C. (2022, 02). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, *55*, 1-50. doi: 10.1007/s10462-022-10144-1

Wu, Y., Jin, Z., Shi, C., Liang, P., & Zhan, T. (2024, 05). Research on the application of deep learning-based bert model in sentiment analysis. *Applied and Computational Engineering*, *71*, 14-20. doi: 10.54254/2755-2721/71/2024MA

Wu, Z., & Ong, D. (2021, 05). Context-guided bert for targeted aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*, 14094-14102. doi: 10.1609/aaai.v35i16.17659

Yadav, K. (2020). A comprehensive survey on aspect based sentiment analysis. *ArXiv*, *abs/2006.04611*. Retrieved from https://api.semanticscholar.org/CorpusID:219530886

Zhang, K., Zhang, K., Zhang, M., Zhao, H., Liu, Q., Wu, W., & Chen, E. (2022, May). Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Findings of the association for computational linguistics: Acl 2022* (pp. 3599–3610). Dublin, Ireland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.findings-acl.285/ doi: 10.18653/v1/2022.findings-acl.285

Appendix A

Appendix

A.1 Gantt Chart

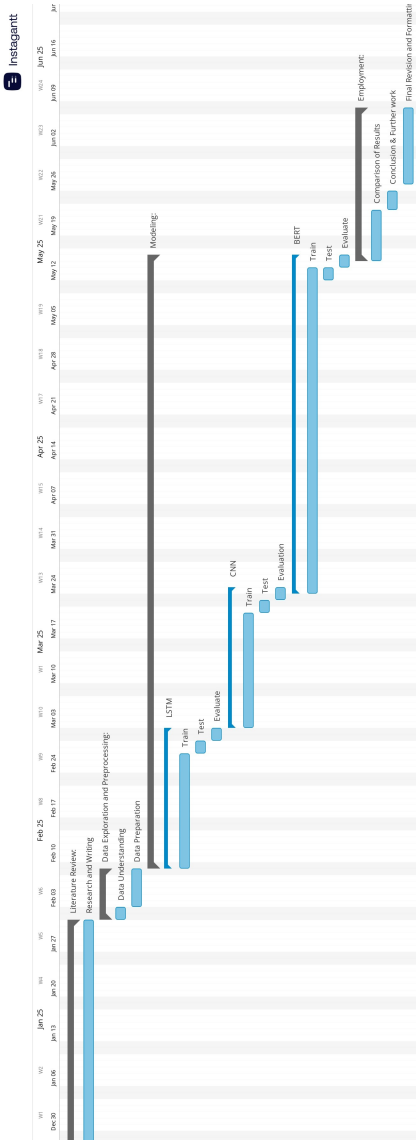

Figure A.1: Gantt Chart

A.2 Word clouds

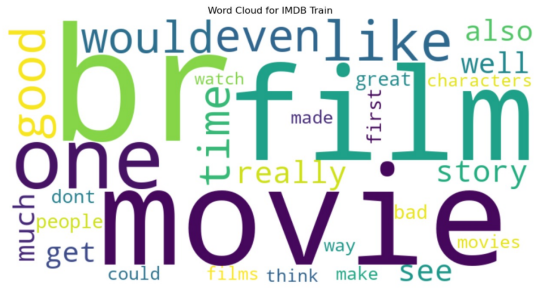

Figure A.2: IMDB Word cloud

Figure A.3: Amazon Word cloud

Figure A.4: T4SA Word cloud

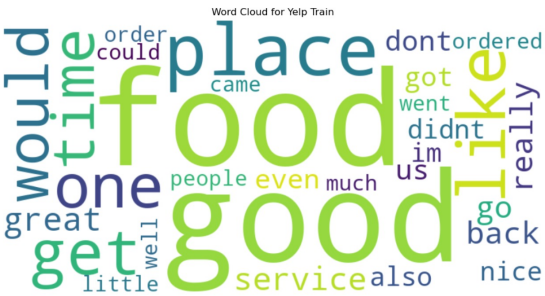

Figure A.5: Yelp Word Cloud

Figure A.6: SemEval Word Cloud

A.3 3-grams

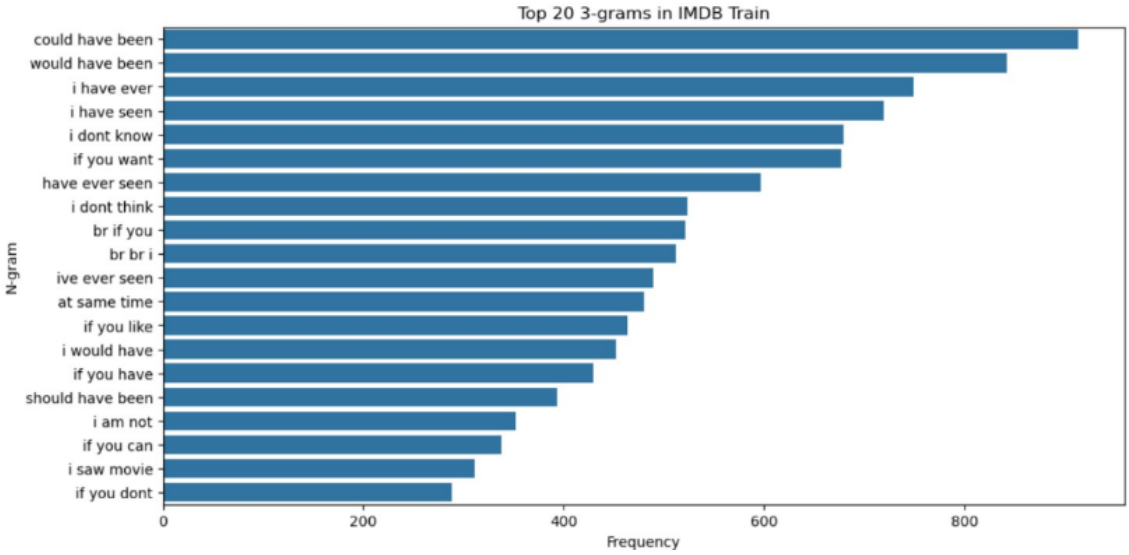

Figure A.7: IMDB 3-grams

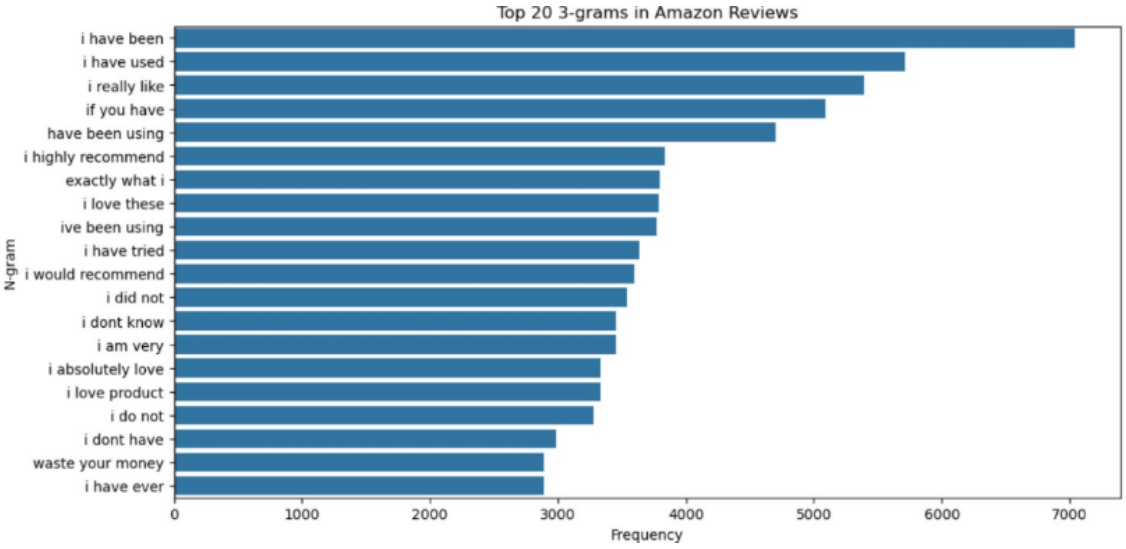

Figure A.8: Amazon 3-grams

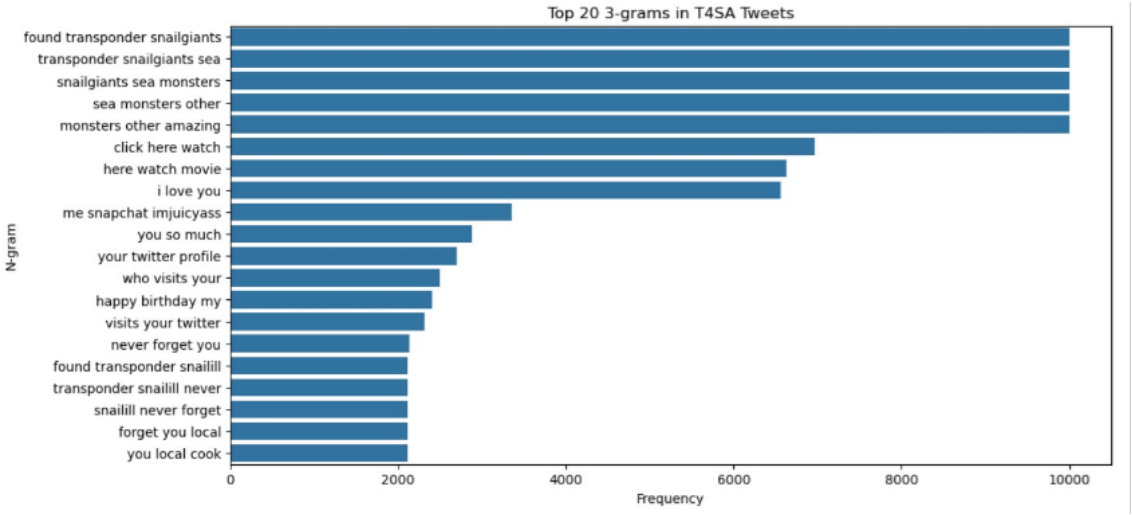

Figure A.9: T4SA 3-grams

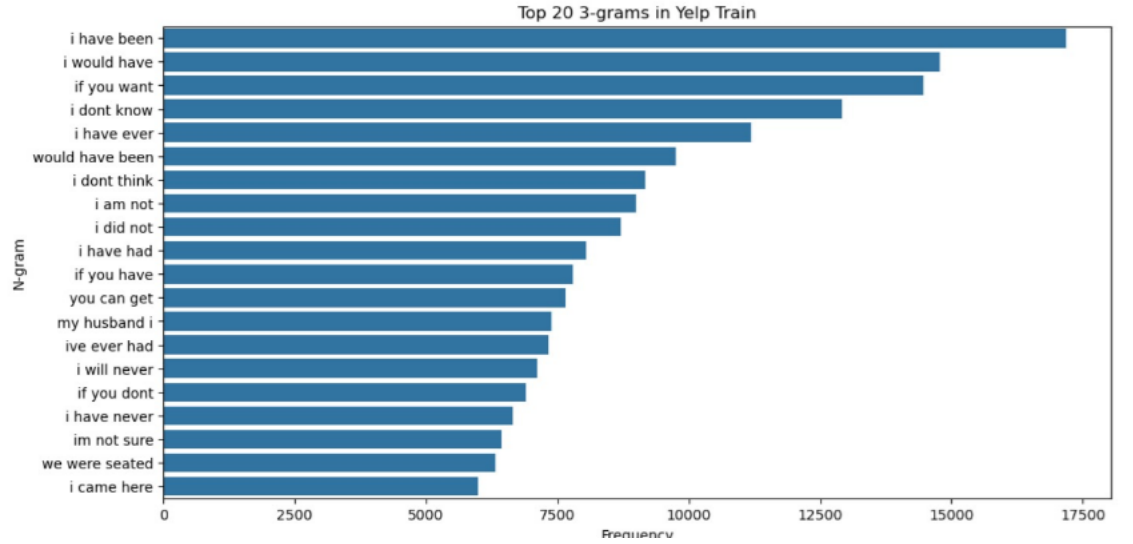

Figure A.10: Yelp 3-grams

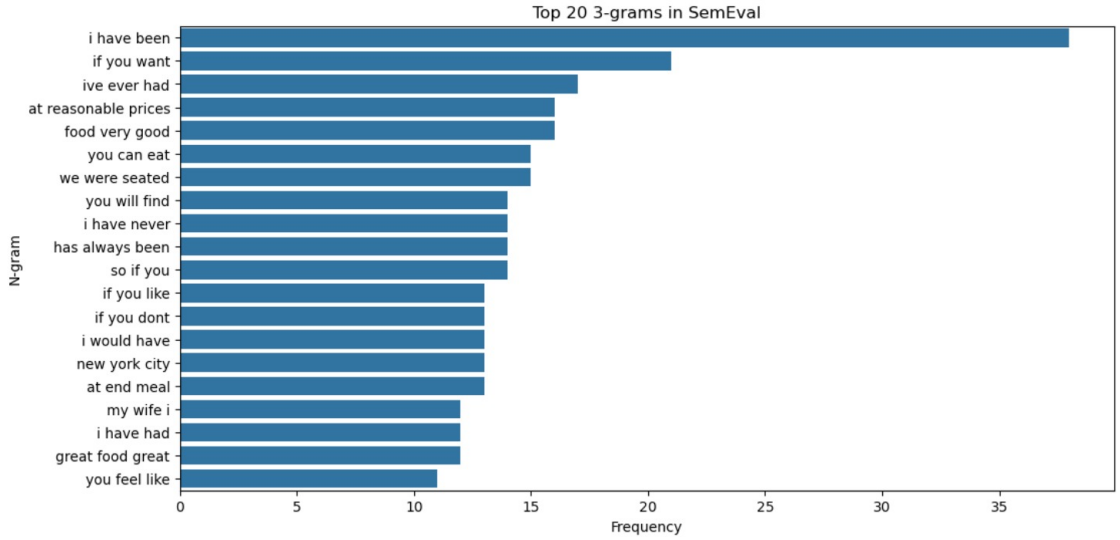

Figure A.11: SemEval 3-grams

A.4 Post-hoc Nemenyi Test P-Value Matrix and Critical Difference Diagram

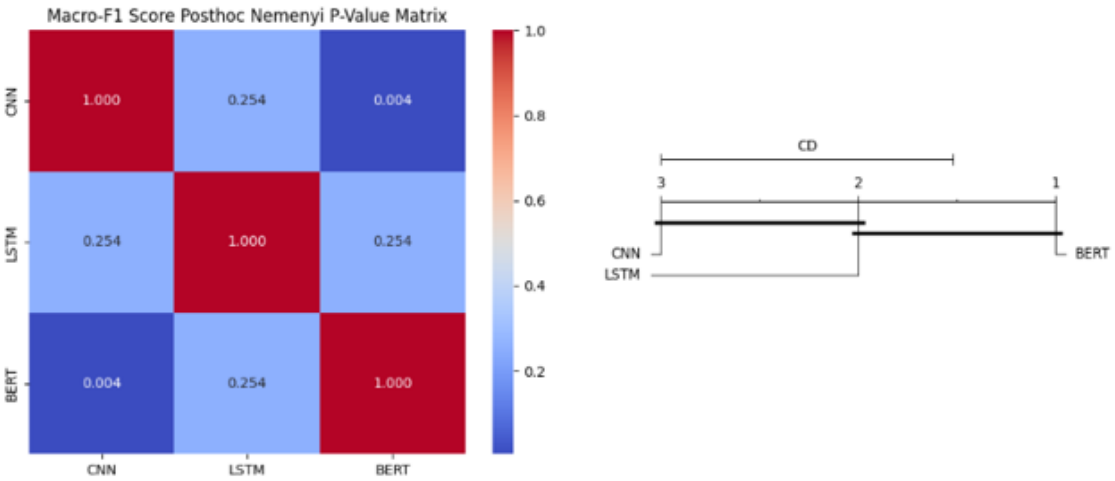

Figure A.12: Macro-F1 Score: Post-hoc Nemenyi Test P-Value Matrix and Critical Difference Diagram

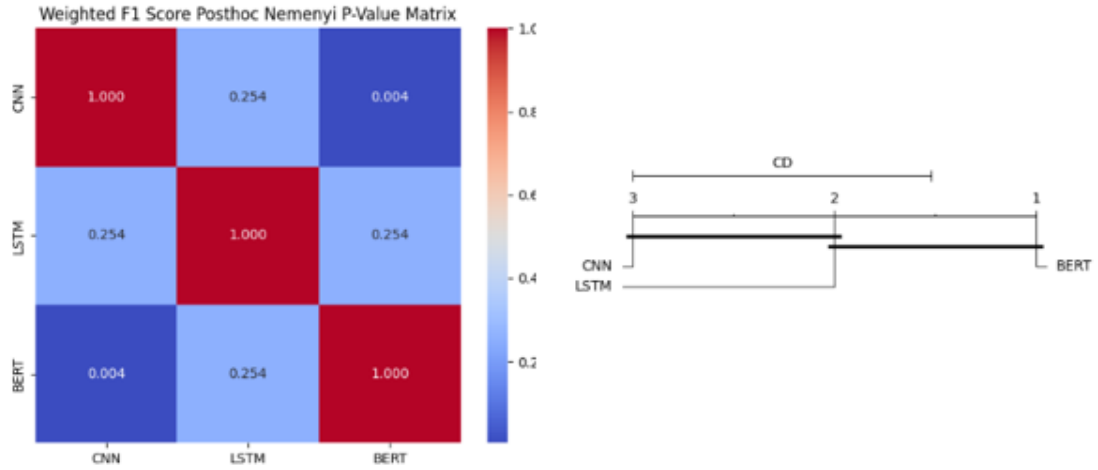

Figure A.13: Weighted-F1 Score: Post-hoc Nemenyi Test P-Value Matrix and Critical Difference Diagram