MASCoT: Multi-aspect Sentiment Analysis with Contrastive learning and Temporal insights

SHREYA KOTHARI
SID: 500594412

Supervisor: Dr. Maryam Khanian Najafabadi

This thesis is submitted in partial fulfillment of
the requirements for the degree of
Bachelor of Advanced Computing (Honours)

School of Computer Science
The University of Sydney
Australia

27 May 2024

Student Plagiarism: Compliance Statement

I Shreya Kothari certify that:

I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);

This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that it is not my own by acknowledging the source of that part or those parts of the work.

Name: Shreya Kothari

Signature: Shreya Kothari **Date**:27-05-2024

Abstract

Aspect-based sentiment analysis (ABSA) is a pivotal task in natural language processing, focusing on identifying the sentiment polarity of specific aspects within a text. Despite significant advancements with pre-trained language models such as BERT, existing methodologies encounter challenges related to multi-aspect interpretability, handling complex interactions, and generalizing across diverse domains.

This study hypothesizes that integrating dynamic modeling techniques with a novel attention mechanism can enhance aspect-specific sentiment discrimination and model interpretability. To test this hypothesis, we developed the Multi-Aspect Sentiment Contrastive Learning (MASCoT) framework. MASCoT employs a novel attention mechanism that highlights aspect importance and generates pairs to improve the model's ability to differentiate between subtle sentiment shifts over time.

Our experimental results show that MASCoT outperforms existing models such as APSCL, AAN, and MTLN, particularly in capturing the intricate relationships between multiple aspects and their associated sentiments. MASCoT demonstrated significant improvements in accuracy and generalizability, particularly on the ACSA MAMS challenge dataset, with a multi-head attention mechanism ensuring precise sentiment capture even with multiple aspects in a single sentence. For instance, MASCoT achieved an accuracy improvement of upto 5% over baseline models and enhanced F1 scores by 4%.

These findings highlight MASCoT's potential to revolutionize real-time sentiment analysis applications, offering enhanced accuracy, interpretability, and generalizability. The integration of contrastive learning and dynamic modeling techniques in MASCoT paves the way for more advanced and reliable sentiment analysis systems, benefiting various domains and improving the understanding of user-generated content in the NLP community.

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Professor Maryam Khanian Najafabadi, for her steadfast guidance, meticulous direction, and passionate support throughout my research journey. Her vast knowledge and perceptive suggestions have been crucial in moulding this work.

I am also immensely grateful to my NLP unit tutor, Henry Weld, for his mentorship and priceless recommendations, which helped me navigate my research on the proper course. His assistance has been vital in overcoming obstacles.

Furthermore, I convey my sincere gratitude to my friends, whose unwavering encouragement and support were essential in helping me reach this milestone. Their consistent check-ins and inspiring words continually motivated me to strive for excellence.

In addition, I would like to acknowledge the invaluable role of various software tools and platforms that have significantly contributed to the completion of this thesis. Claude.ai and ChatGPT served as intelligent sounding boards, offering valuable suggestions for improving the flow and structure of the thesis. Grammarly helped in identifying and correcting grammatical errors, thereby enhancing the overall quality of the writing. Draw.io enabled the creation of clear and professional illustrations that effectively conveyed complex concepts. Zotero, a comprehensive reference management system, streamlined the process of organizing and citing sources. Finally, Weights and Biases facilitated the efficient tracking and optimization of hyperparameters. The collective impact of these tools cannot be overstated, and their seamless integration into the research workflow has greatly enhanced the quality and efficiency of this thesis.

Lastly, I am profoundly grateful to my parents and my siblings for their constant support and encouragement, both emotionally and financially. Their belief in me has been a source of strength and motivation, making it possible to navigate through the challenges encountered during this project.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Contrastive Learning

Recent advancements in artificial intelligence (AI) have significantly transformed various domains, particularly in areas like computer vision and natural language processing (NLP). Leveraging vast datasets available from the internet, researchers have made notable strides in developing sophisticated AI models capable of processing and analyzing complex data. However, despite these advancements, current AI systems often fall short in terms of learning flexibility and adaptability compared to human learning.

Human learning is characterized by the ability to understand and capture nuanced discrepancies between language and true sentiments through context and experience. We, as humans, can easily discern the subtle differences in emotions conveyed through text or speech, even when the surface-level words may not explicitly reflect those sentiments. This ability to comprehend the underlying meaning and context is crucial for effective communication and understanding.

To bridge this gap and enable AI systems to learn in a more human-like manner, there is a growing need for advanced techniques that can capture the nuances and subtleties of language. One such technique that has gained significant attention in recent years is contrastive learning.

Contrastive learning, rooted in metric learning, focuses on learning representations by contrasting positive pairs (similar examples) against negative pairs (dissimilar examples). Contrastive learning has shown remarkable success in various domains, particularly in computer vision tasks such as image classification, object detection, and segmentation [2]. The ability of contrastive learning to learn transferable visual representations has led to significant improvements in performance, often surpassing supervised learning approaches [3]. These successes have inspired researchers to explore the application of contrastive learning in other domains, like natural language processing (NLP).

Contrastive learning has emerged as a powerful technique in NLP, offering a novel approach to learning robust and meaningful representations of textual data. By learning to distinguish between similar and dissimilar examples, contrastive learning models can capture the underlying semantic structure of the data and develop a deeper understanding of the relationships between different text samples. This is particularly valuable in tasks where understanding the nuances of language is crucial, such as sentiment analysis, text summarization, and question answering.

However, applying contrastive learning to NLP tasks presents unique challenges compared to its application in computer vision. Text data is sequential and variable in length, requiring specialized techniques to handle these properties effectively. Additionally, the notion of similarity between text examples can be more nuanced and context-dependent, complicating the selection of positive and negative pairs for contrastive learning.

Despite these challenges, the potential benefits of contrastive learning in NLP are significant. By learning rich and transferable representations of textual data, contrastive learning can improve the performance of NLP models across a wide range of tasks and reduce the reliance on large-scale annotated datasets. In this thesis, we will be employing the contrastive learning framework on a crucial NLP task: Multi-Aspect Based Sentiment Analysis.

1.2 Aspect-Based Sentiment Analysis

Sentiment analysis is a fundamental task in natural language processing that aims to determine the overall sentiment expressed in a piece of text. It classifies the sentiment as positive, negative, or neutral based on the emotional tone and opinions conveyed in the text. For example, let's consider the sentence "The food at this restaurant was delicious, but the service was terrible." In traditional sentiment analysis, this review would likely be classified as having a mixed or neutral sentiment. The positive sentiment expressed towards the food ("delicious") is offset by the negative sentiment towards the service ("terrible"). The overall sentiment score might not clearly indicate whether the reviewer had a positive or negative experience at the restaurant.

This limitation can lead to an oversimplification of sentiment understanding and hinder the ability to extract actionable insights. To address this issue, a more fine-grained approach called aspect-based sentiment analysis (ABSA) has emerged, which focuses on identifying and extracting the sentiment expressed towards specific aspects or features within a text.

1.2.1 Importance of Aspect-Based Sentiment Analysis

Aspect-Based Sentiment Analysis (ABSA) holds significant importance in a variety of applications, especially in domains where understanding fine-grained sentiment towards specific features is critical. ABSA addresses this limitation faced by traditional Sentiment Analysis by breaking down the sentence to the level of individual aspects, and providing a more granular and detailed understanding of sentiment.

For instance, for the restaurant review: "The food at this restaurant was delicious, but the service was terrible.", ABSA would break down the sentiment analysis into specific aspects mentioned in the text, such as “food” and “service”. For the aspect of “food," ABSA would identify the sentiment as positive based on the word “delicious” and negative for “the service” based on the word “terrible”. By dissecting these sentiments, businesses can pinpoint exactly which features of their products or services are being praised or criticized.

Moreover, the technology sector also leverages ABSA for software and product reviews. Developers and companies can gain detailed feedback on various aspects of their software, such as user interface, functionality, and performance. This granular feedback is crucial for iterative development and enhancing user experience. For instance, while a software application may be praised for its functionality, ABSA can identify recurring issues with its user interface, guiding developers on where to focus their efforts for the next update.

The importance of ABSA extends to academic research and public policy as well. Researchers can utilize ABSA to analyze public opinion on various social, political, and economic issues, providing insights that can inform policy decisions. For instance, understanding public sentiment towards different aspects of healthcare policies can help policymakers address specific concerns and improve the overall effectiveness of their initiatives.

In conclusion, ABSA provides a nuanced and detailed understanding of sentiment that is essential for various industries and applications. By breaking down sentiment analysis to the aspect level, ABSA enables businesses and organizations to gain actionable insights, improve customer satisfaction, make data-driven decisions, and enhance overall performance. The ability to dissect and analyze sentiment towards specific aspects is what makes ABSA a powerful tool in today’s data-driven world.

1.3 Challenges of ABSA

Despite the significant progress made in ABSA research, several challenges remain that hinder the development of highly accurate and reliable ABSA systems. These challenges arise from the complexity of natural language, the nuances of sentiment expression, and the limitations of current approaches. One of the primary challenges in ABSA is the accurate identification and extraction of aspects from text data. Aspects can be expressed implicitly or explicitly, and their mention can be context-dependent [4]. For example, in the sentence "The battery life is great, but the camera quality is poor," the aspects "battery life" and "camera quality" are explicitly mentioned. However, in the sentence "The phone fits perfectly in my hand," the aspect "size" is implicitly referred to. Identifying such implicit aspects requires a deep understanding of the context and domain knowledge.

Another challenge lies in the association of sentiments with their corresponding aspects. Sentiments can be expressed at different granularities, ranging from the overall sentiment of a document to the sentiment towards specific aspects within a sentence [5]. Moreover, a single sentence may contain multiple aspects with varying sentiments. For example, in the sentence "The food was delicious, but the service was terrible," the sentiment towards the aspect "food" is positive, while the sentiment towards the aspect "service" is negative. Accurately linking sentiments to their respective aspects is crucial for effective ABSA.

The presence of complex linguistic structures, such as negation, sarcasm, and conditional statements, further complicates the task of ABSA [6]. These linguistic phenomena can alter the sentiment polarity associated with aspects and require sophisticated techniques to handle them effectively. Additionally, the use of domain-specific terminology and colloquial expressions can vary across different domains, making it challenging to develop ABSA models that generalize well to unseen domains.

Moreover, ABSA often suffers from data scarcity and class imbalance issues. Annotating large volumes of text data with aspect-level sentiment labels is time-consuming and labor-intensive, leading to limited availability of high-quality labelled datasets [7]. Furthermore, the distribution of sentiment classes may be skewed, with some sentiment polarities being more prevalent than others. This class imbalance can bias the training of ABSA models and affect their performance on minority classes. Addressing these challenges requires the development of advanced techniques that can effectively capture the nuances of aspect-level sentiments, handle complex linguistic structures, and overcome data scarcity and class imbalance issues. This has led to the exploration of various approaches, such as deep learning models, attention mechanisms, and transfer learning, to improve the performance of ABSA systems [8].

1.4 Our Contributions

The primary objective of this thesis is to address the challenges posed by multi-aspect sentiment analysis and propose a novel framework, MASCoT (Multi-aspect Sentiment Analysis with Contrastive learning and Temporal insights), which leverages contrastive learning and temporal information to enhance the performance of ABSA models. The proposed framework aims to capture the nuances of aspect-level sentiments, handle multiple aspects within a single sentence, and incorporate temporal dynamics to provide a comprehensive understanding of sentiment evolution over time.

The motivation behind this research stems from the limitations of existing ABSA approaches, which often struggle to effectively handle sentences containing multiple aspects with potentially conflicting sentiment polarities [4]. Moreover, the temporal dimension of sentiment expression is often overlooked, resulting in a static representation of sentiments that fails to capture the dynamic nature of opinions over time [5]. By addressing these gaps, MASCoT seeks to provide a more accurate and insightful analysis of aspect-based sentiments, enabling businesses, organizations, and researchers to make informed decisions based on a deeper understanding of public opinion and customer feedback.

The main contributions of this thesis are as follows:

(1) Proposing MASCoT, a novel framework for multi-aspect sentiment analysis that integrates contrastive learning and temporal insights.
(2) Introducing an original multi-head attention mechanism to effectively capture aspect-level sentiment nuances by incorporating detailed aspect information into the model.
(3) Implementing a temporal window splitting technique to capture the temporal context of sentiments, further improving the model's understanding of aspect-specific sentiments.
(4) Developing a contrastive learning approach that generates positive and negative pairs based on sentiment polarity and aspect similarity, enhancing the model's ability to distinguish between different sentiment contexts.
(5) Implementing Focal Loss within the contrastive framework to overcome issue of class imbalance and unlabelled data.

(6) Conducting extensive experiments on the MAMS dataset to demonstrate the effectiveness of MASCoT in handling multi-aspect and temporally evolving sentiments.

(7) Link for the code can be found at the following GitHub Repository.

1.5 Thesis Outline

This thesis is structured into nine chapters, each addressing a key aspect of our research on aspect-based sentiment analysis (ABSA) using contrastive learning and temporal insights.

- Chapter 1 introduces the context, motivation, and challenges of ABSA, along with our main contributions and thesis structure.
- Chapter 2 provides background on ABSA approaches, including frequency-based, syntax-based, lexicon-based, machine learning-based, and deep learning-based methods. It also covers relevant concepts such as embeddings, pre-trained language models, attention mechanisms, contrastive learning, and temporal analysis.
- Chapter 3 presents a literature review, identifying knowledge gaps and discussing our research on improving model interpretability, temporal analysis, handling multiple aspects with conflicting sentiments, enhancing aspect-specific sentiment discrimination, and improving generalizability.
- Chapter 4 introduces our Multi-Aspect Sentiment Contrastive Learning (MASCoT) framework, detailing the input processing, contextual embeddings retrieval, and contrastive learning modules.
- Chapter 5 focuses on experimental results, including setup, hyperparameter tuning, validation accuracy, training and validation loss, and final test evaluation.
- Chapter 6 provides a discussion of classification results, an ablation study, model design and architecture, window size and fine-tuning, resource constraints and optimization, relation to current knowledge, and implementation challenges and future directions.
- Chapter 7 addresses threats to validity, including internal, external, construct, and conclusion validity.
- Chapter 8 discusses limitations and future research directions, such as incorporating domain-specific knowledge, handling sarcasm and irony, and adapting to multilingual and cross-domain ABSA tasks.

- Chapter 9 concludes the thesis, summarizing key contributions, insights, and the potential impact of our research on real-world applications.

CHAPTER 2

Background

Sentiment analysis, often termed as opinion mining or opinion analysis, stems from the innate curiosity to comprehend the sentiments and viewpoints of others. It has assumed a position of paramount importance in the modern world and has not only captured the attention of researchers but has also resonated with businesses, governments, and various organizations [9]. Whether it involves an individual making a choice regarding a travel destination or a company refining the design of one of its products, the collective opinion holds a decisive sway over the outcomes [10]. This heightened interest can largely be attributed to the internet's transformation into the primary source of global information.

In the landscape of sentiment analysis, two fundamental approaches have long been employed to understand the sentiments expressed in text: sentence-level sentiment analysis and phrase-level sentiment analysis [11].

Sentence-level analysis entails the examination of individual sentences to determine their corresponding polarity; it proves particularly valuable when a document exhibits a diverse and mixed array of sentiments [12]. Previous efforts in sentence-level analysis have primarily concentrated on identifying subjective sentences. Nevertheless, tackling more intricate tasks, such as dealing with conditional, comparative or ambiguous statements [13], underscores the significance of sentence-level sentiment analysis in such contexts.

Alternatively, Phrase-level sentiment analysis involves the mining of opinion words at the phrase level, followed by classification. Each phrase may encompass multiple aspects or focus on a single aspect, making it valuable for product reviews where a single aspect can be conveyed within a phrase [14].

While these traditional approaches have laid the foundation for sentiment analysis, they have been increasingly superseded by a more sophisticated technique known as aspect-based sentiment analysis

(ABSA). ABSA represents a paradigm shift in sentiment analysis, offering a finer-grained and more context-aware understanding of sentiment.

Aspect-Based Sentiment Analysis

ABSA focuses on dissecting text into distinct aspects or facets, such as specific product features or aspects of a service and assesses sentiment for each aspect independently. This approach allows for a nuanced examination of sentiment within longer texts, such as product reviews or social media comments, where different aspects may convey varying sentiments. ABSA offers greater precision in capturing the intricacies of sentiment expression, making it a preferred choice in modern sentiment analysis research.

ABSA encompasses three essential subtasks, as outlined by Pontiki et al. [15]:

(1) Opinion Target Extraction (OTE): Identifies and extracts aspect terms, representing entities or attributes that are the subject of opinions or sentiments in the text. For example, in the sentence "I love the sushi here," the aspect term would be "sushi."
(2) Aspect Category Detection (ACD): Associates the identified aspect terms with predefined categories, which typically include entities and attributes. In the same sentence, "sushi" could be categorized as an entity within the "food" category and as an attribute indicating "quality."
(3) Sentiment Polarity (SP): Determines the sentiment polarity associated with the identified aspect terms, classifying sentiments as either "positive" or "negative." For instance, in the sentence "The sushi is great," the sentiment polarity associated with "sushi" is "positive."

The three ABSA subtasks are interrelated and rely on one another. Accurate OTE is crucial for successful ACD and SP, as the recognition of aspect categories and sentiment polarity depends on correctly identifying the opinion target. The correlation between ACD and SP is strong, as they both involve aspects and sentiment. As shown in Figure 1, the correlation between ACD and SP is strong, as they both involve aspects, such as "sushi," and sentiment, like "great."

Furthermore, OTE primarily focuses on explicit opinion targets mentioned in the text, while ACD extends its scope to cover both explicit and implicit aspects inferred from the context. Implicit opinion targets, like "HP" in the sentence "My HP is very heavy," provide valuable information for understanding sentiment, even though they are not explicitly stated. This distinction enhances the depth of sentiment analysis by addressing both explicit and implicit aspects.

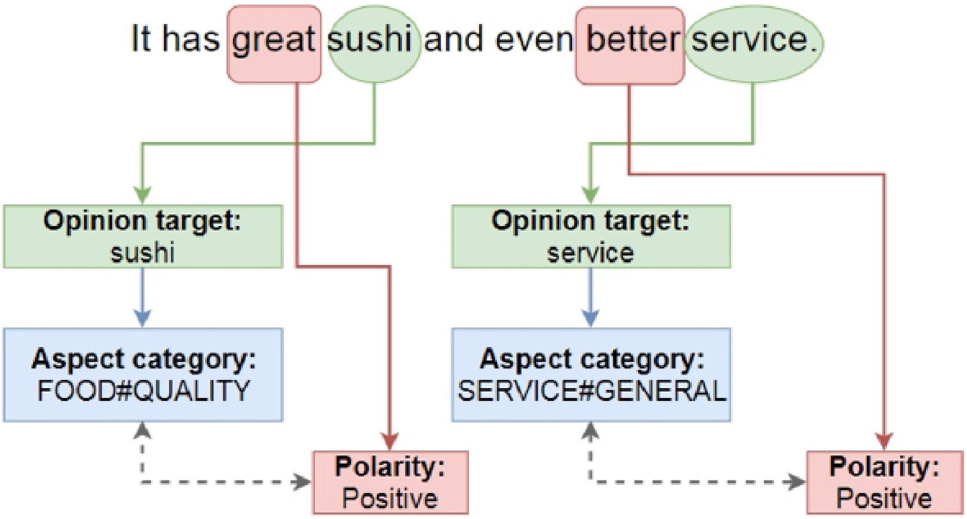

FIGURE 2.1: Illustrates the three core tasks of Aspect-Based Sentiment Analysis (ABSA) using a sentence from the SemEval ABSA dataset 2016.

ABSA dissects text into specific aspects and assesses sentiment for each aspect independently, allowing for a nuanced examination of sentiment within longer texts. This ability to focus on specific aspects and their associated sentiments has proven to be more relevant and insightful, especially in the context of consumer feedback, product improvement, and decision-making processes.

2.1 Approaches to Aspect-Based Sentiment Analysis (ABSA)

Aspect-based sentiment analysis (ABSA) has seen various approaches and methods over the years, addressing the challenge of extracting aspects from text and understanding their sentiment. These methods can be broadly categorized into frequency-based, syntax-based, lexicon-based, machine learning-based, and deep learning-based approaches.

2.1.1 Frequency-Based Approaches

In the early days of aspect term extraction, researchers utilized frequency-based approaches. The fundamental idea was straightforward: if a word or phrase appeared frequently in the context of product or service reviews, it likely denoted an aspect of significance [16]. However, frequency-based methods faced challenges, particularly with multi-word aspects and capturing intricate relationships between words [17]. They often relied on predefined rules and heuristics, making them less adaptable to diverse domains and textual genres.

Similarity-based approaches emerged as an enhancement, relying on co-occurrence statistics to identify aspects. Words frequently appearing together in sentences or reviews were considered related. However, these methods also struggled with multi-word aspects and the complexity of user-generated content.

Despite their foundational role, frequency-based approaches were limited by their inability to handle complex grammatical structures and multi-word aspects, prompting researchers to seek more advanced techniques for aspect term extraction.

2.1.2 Syntax-Based Approaches

Syntax-based approaches emerged to capture underlying syntactical dependencies within text, enhancing aspect term extraction. These techniques leveraged grammatical structures and relationships embedded in the text.

(1) Grammar and Dependency-Based Models: These models incorporated grammatical structures and dependencies to capture syntactical relationships between words in a sentence. For instance, Sun et al. [18] employed dependency parsing to identify aspects by exploring word relationships in customer reviews.
(2) Localizing Aspect Terms: Some syntax-based approaches treated each sentence as a separate document, allowing models to associate aspects with specific sentences. This technique was used by Titov and McDonald [19] to recognize that aspects in one sentence might not have the same relevance in another.

While syntax-based approaches made strides in aspect term extraction, they struggled with informal or grammatically incorrect text and required well-structured sentences.

2.1.3 Lexicon-Based Approaches

Lexicon-based methods emerged as a prominent approach for deciphering sentiment polarity. These methods recognized the sentiment of words in context, surpassing frequency-based methods by focusing on qualitative aspects of sentiments. Lexicon-based approaches could differentiate between synonymous words used in different sentiment contexts [20].

One of the fundamental advantages of lexicon-based methods is their transparency and interpretability, providing a clear mapping of words to sentiment categories [21]. Advancements in lexicon-based approaches included techniques for automatic lexicon expansion tailored to domain-specific sentiment analysis [22].

Despite their strengths, lexicon-based approaches are dependent on the quality and comprehensiveness of sentiment lexicons, which can restrict their performance in specialized or evolving vocabularies. They also struggle with nuanced sentiment expressions involving sarcasm, irony, or complex linguistic structures [23].

2.1.4 Machine Learning-Based Approaches

Machine learning classifiers have emerged as a more sophisticated and versatile choice for Aspect-Based Sentiment Analysis (ABSA), offering improved accuracy and adaptability compared to lexicon-based approaches. The transition to machine learning methods has been primarily driven by the need to capture the context and semantics of language more effectively, which is crucial in ABSA tasks [24, 25].

Support Vector Machines (SVM) is a prominent machine learning classifier used in ABSA. SVMs have effectively handled high-dimensional feature spaces, making them particularly suitable for ABSA tasks [26]. By maximizing the margin between different classes, SVMs can effectively classify sentiment polarities associated with specific aspects [27].

Another widely used machine learning classifier is Logistic Regression. Logistic Regression offers a straightforward yet reliable approach to ABSA [28]. It models the probability of an instance belonging to a particular sentiment class based on a linear combination of input features. The simplicity and interpretability of Logistic Regression make it an attractive choice for sentiment classification tasks [29].

Decision Trees have also found application in ABSA due to their ease of interpretation [30]. Decision Trees recursively partition the feature space based on a set of rules, creating a tree-like structure. Each internal node represents a decision based on a specific feature, while the leaf nodes represent the sentiment classes. The transparency of Decision Trees allows for a clear understanding of the decision-making process, which can be valuable in analyzing sentiment patterns [31].

Maximum Entropy Models, also known as Logistic Regression with regularization, have been employed in ABSA to model complex relationships while balancing performance and computational cost [32]. These models estimate the conditional probability distribution of sentiment classes given the input features, by maximizing the entropy of the distribution subject to certain constraints. Maximum Entropy

Models are particularly useful when dealing with a large number of features and can effectively handle feature interactions [33].

Advantages of Machine Learning-Based Approaches

Machine learning classifiers offer several advantages in the context of ABSA. Firstly, they excel at understanding the context and semantics of language, which is crucial in capturing the sentiment expressed towards specific aspects. By considering the surrounding words and their relationships, machine learning classifiers can disambiguate the sentiment of a word based on its context [34].

Secondly, machine learning classifiers are highly adaptable. They can be trained on domain-specific datasets, allowing them to learn the nuances and terminology specific to a particular industry or domain. This adaptability enables machine learning classifiers to perform well across various domains, such as product reviews, social media posts, or customer feedback [35].

Moreover, machine learning classifiers offer flexibility in terms of feature representation. They can handle a wide range of features, including unigrams, bigrams, and more complex linguistic features such as part-of-speech tags or dependency relations [36]. This feature flexibility allows machine learning classifiers to capture rich syntactic and semantic information from the text, enhancing their ability to accurately classify sentiment polarities.

Machine learning classifiers are also capable of handling negations and modifiers, which are common in sentiment expressions. By considering the presence of negation words or sentiment modifiers, these classifiers can accurately determine the sentiment polarity of an aspect, even in the presence of complex linguistic structures [37].

Furthermore, machine learning classifiers have shown effectiveness in identifying sarcasm and figurative language in text. Sarcasm and figurative expressions often pose challenges for sentiment analysis, as they can reverse the apparent sentiment polarity. By learning patterns and indicators of sarcasm or figurative language from labeled training data, machine learning classifiers can improve their accuracy in handling such cases [38].

Disadvantages of Machine Learning-Based Approaches

However, machine learning classifiers also have certain limitations. One major limitation is their heavy reliance on labeled training data. Obtaining high-quality labeled data, especially for fine-grained aspects, can be time-consuming and resource-intensive. The performance of machine learning classifiers is highly dependent on the quality and quantity of the labeled data used for training [10].

Another limitation is the potential for complexity and overfitting. As the number of features and the complexity of the model increase, machine learning classifiers can become overly complex, leading to overfitting. Overfitting occurs when the model learns to fit the noise or peculiarities of the training data, rather than capturing the underlying patterns. This can result in poor generalization to unseen data, especially when working with smaller datasets [39].

Lastly, some machine learning classifiers, such as Support Vector Machines, can be computationally intensive, requiring significant computational resources and longer training times. This can be a practical limitation when dealing with large-scale ABSA tasks or real-time sentiment analysis applications [40].

Despite these limitations, machine learning classifiers have proven to be valuable tools in ABSA, offering improved accuracy, adaptability, and the ability to capture complex sentiment patterns. By leveraging the power of machine learning, researchers and practitioners can develop more effective and nuanced sentiment analysis models, enabling better understanding and insights from textual data.

2.1.5 Deep Learning-Based Models

The transition from traditional machine learning methods to Deep Learning (DL) for Aspect-Based Sentiment Analysis (ABSA) has been driven by the need for more effective and efficient techniques. Earlier ABSA approaches primarily relied on machine learning algorithms and linguistic features, which, while effective, had limitations such as the requirement for extensive domain-specific datasets and manual annotation [41]. This led to the exploration of DL methods for ABSA tasks. Deep Learning is a machine learning paradigm that involves learning data representations through artificial neural networks with multiple layers. One of the central DL concepts is Deep Neural Networks (DNNs), which are artificial neural networks comprising numerous layers of interconnected units or neurons, as shown in Fig. 2.2. DNNs process input data through these layers, each layer extracting progressively abstract and complex features. The key to the success of DNNs is their ability to automatically learn relevant features from data, reducing the need for manual feature engineering [42].

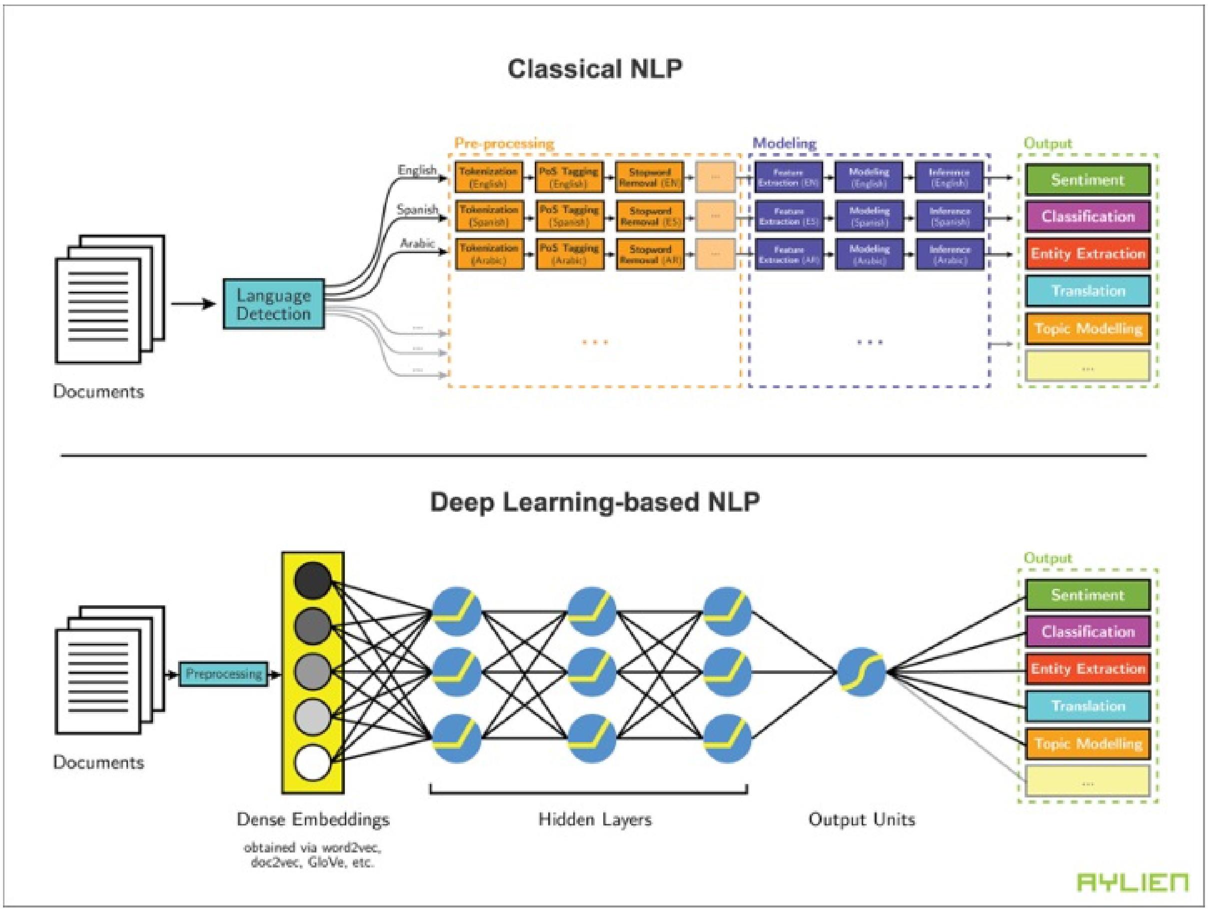

FIGURE 2.2: Comparison of classic machine learning and deep learning processes for NLP. Deep learning architecture is characterized by dense embeddings and hidden layers (adapted from [1].

Advantages of Deep Learning-Based Models

It is important to note that while traditional machine learning methods are still relevant in some applications, the transition to DL has provided new opportunities for ABSA research by addressing some of the limitations of previous approaches. They excel in feature engineering, automating the extraction of pertinent features from diverse and multilingual data, reducing the reliance on predefined features [42]. Moreover, DL models prioritize generalization, capturing intricate patterns and relationships in the data to handle a broad spectrum of aspects and sentiments, in contrast to traditional models limited by specific features [43]. Additionally, they excel in semantic comprehension, proficiently recognizing sentiment expressions even in contexts with negations, sarcasm, and context-dependent sentiments, a critical skill in ABSA where context shapes sentiment [44]. Lastly, these models facilitate fine-grained analysis, enabling aspect-level sentiment identification by hierarchically processing text, surpassing traditional models in granularity [45]. Their adaptability to diverse domains, necessitating less domain-specific data and enabling knowledge transfer, further underscores their relevance for practical ABSA applications [46].

Disadvantages of Deep Learning-Based Models

Deep Learning (DL) methods hold promise in Aspect-Based Sentiment Analysis (ABSA), but challenges and limitations persist. Some studies report mixed results, where DL models underperform compared to traditional methods [47, 48]. Domain adaptation poses a significant challenge, given that word sentiment can vary greatly with context. Models excelling in one domain may not generalize effectively to others [49]. Performance disparities across domains further complicate ABSA, influenced by varying aspect prevalence and format [50, 51]. The lack of domain-specific lexicons and the need for opinion-based pre-training to enhance domain adaptation present additional hurdles [52, 53]. In summary, while DL methods hold potential for ABSA, addressing these challenges, including variable performance, domain adaptation, and domain-specific resources, is essential to maximize their effectiveness. Ongoing research seeks to develop strategies for mitigating these issues and advancing DL's application in ABSA.

2.2 Relevant Concepts in ABSA

2.2.1 Embeddings

Embeddings are dense vector representations of words or phrases that capture their meanings based on the context in which they appear. Traditional embeddings, like Word2Vec or GloVe, represent words as fixed vectors irrespective of their context in a sentence [54, 55]. These embeddings are trained on large corpora to capture semantic similarities between words based on their co-occurrence patterns.

Contextual embeddings, on the other hand, provide dynamic representations that change depending on the context of the surrounding words. Models like BERT and RoBERTa generate contextual embeddings by considering the entire sentence, allowing them to capture the nuances of word meanings in different contexts [56, 57]. For example, the word "bank" will have different embeddings in the sentences "He sat on the river bank" and "She deposited money in the bank".

These contextual embeddings are crucial for advanced NLP tasks as they enable models to understand and process the complexities of human language more effectively. In the MASCoT framework, contextual embeddings are used to generate aspect-specific representations, enhancing the model's ability to discern subtle differences in sentiment across various aspects.

The choice of embedding technique can significantly impact the performance of ABSA models. While traditional embeddings like Word2Vec and GloVe have been widely used in the past, recent research has shown that contextual embeddings, such as those generated by BERT and its variants, consistently outperform their non-contextual counterparts [58, 59]. This is because contextual embeddings can better capture the complex semantic relationships between words and their context, which is essential for understanding the sentiment expressed towards specific aspects in a sentence.

Moreover, the quality of the embeddings is also influenced by the size and diversity of the training corpus. Embeddings trained on larger and more diverse datasets tend to be more robust and generalizable [54, 55]. However, domain-specific embeddings, trained on a corpus relevant to the target domain (e.g., restaurant reviews for ABSA in the restaurant domain), can often lead to better performance than generic embeddings [59, 60].

In summary, the choice of embedding technique is a crucial consideration in the development of ABSA models. Contextual embeddings, particularly those generated by state-of-the-art models like BERT and RoBERTa, have become the preferred choice due to their superior ability to capture semantic relationships and context. The MASCoT framework leverages these advancements in embedding techniques to generate high-quality, aspect-specific representations that enhance the model's sentiment analysis performance.

2.2.2 Pre-trained Language Models (e.g., BERT, RoBERTa)

Pre-trained language models have revolutionized natural language processing (NLP) by providing powerful tools for understanding and generating human language. Among these, BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa (Robustly optimized BERT approach) are two of the most influential models.

BERT, introduced by Devlin et al. [56], uses a transformer-based architecture that is bidirectional, meaning it considers the context from both the left and right sides of a given word. This bidirectional approach allows BERT to capture nuanced meanings and relationships in text [56]. BERT is pre-trained on a vast corpus of text using two unsupervised tasks: masked language modelling (MLM) and next sentence prediction (NSP). Once pre-trained, BERT can be fine-tuned on a specific task with relatively little additional data, making it highly adaptable for various NLP applications, including sentiment analysis [58, 61].

RoBERTa, developed by Liu et al. [57], builds upon BERT by optimizing the pre-training process. It removes the NSP task and focuses solely on MLM, increasing the size of the training data and the duration of training [57]. RoBERTa also uses dynamic masking and larger batch sizes, resulting in a more robust model that outperforms BERT on several NLP benchmarks [57, 62].

These pre-trained models serve as the backbone for many modern NLP tasks, providing a strong foundation for further fine-tuning on specific applications, such as aspect-based sentiment analysis (ABSA) [63, 59].

The success of pre-trained language models like BERT and RoBERTa can be attributed to several factors. Firstly, these models are trained on massive amounts of unlabeled text data, allowing them to learn rich representations of language that capture complex semantic and syntactic relationships [56, 57]. This pre-training process enables the models to develop a deep understanding of language structure and meaning, which can then be leveraged for various downstream tasks.

Secondly, the transformer architecture employed by these models allows for efficient parallel processing and enables the capture of long-range dependencies in text [64]. This is particularly important for ABSA, where the sentiment expressed towards an aspect may be influenced by words that are not in its immediate vicinity.

Furthermore, the fine-tuning process allows these pre-trained models to be easily adapted to specific tasks and domains with relatively little labeled data [58]. This is a significant advantage over traditional machine learning approaches, which often require large amounts of labeled data for each task and domain.

However, it is important to note that while pre-trained language models have achieved state-of-the-art performance on many NLP tasks, they are not without limitations. One challenge is the computational resources required to train and fine-tune these large models, which can be prohibitive for some organizations or researchers [65]. Additionally, there are concerns about the potential for these models to perpetuate biases present in their training data, which may lead to unfair or discriminatory outcomes if not properly addressed [66].

Despite these challenges, pre-trained language models like BERT and RoBERTa have become essential tools in the NLP toolkit, and their impact on ABSA has been significant. The MASCoT framework leverages these powerful models to generate rich, context-aware representations of text, enabling more accurate and nuanced sentiment analysis at the aspect level.

2.2.3 Attention Mechanisms

Attention mechanisms have become a cornerstone in NLP, enabling models to focus on relevant parts of the input text when making predictions. Introduced by Bahdanau et al. [67] for neural machine translation, attention mechanisms allow models to weigh the importance of different words in a sentence dynamically [67].

In the context of sentiment analysis, attention mechanisms help models to identify which parts of the text contribute most to the sentiment of a particular aspect. This is crucial for ABSA, where different aspects of the same sentence may have different sentiments. For example, in the sentence "The food was great, but the service was terrible," an attention mechanism can help the model focus on "food" when determining the sentiment of the first aspect and "service" for the second aspect [68, 69].

Multi-head attention, a key component of the transformer architecture, extends this idea by allowing the model to attend to different parts of the sentence from multiple perspectives simultaneously. This improves the model's ability to capture complex relationships and dependencies in the text [64, 70].

The incorporation of attention mechanisms has led to significant improvements in the performance of ABSA models. By enabling models to focus on the most relevant parts of the text for each aspect, attention mechanisms help to reduce noise and improve the accuracy of sentiment predictions [68, 71].

Various attention mechanisms have been proposed for ABSA, each with its own strengths and weaknesses. Some common approaches include:

1. Aspect-aware attention: This approach involves generating aspect-specific attention weights, allowing the model to focus on the parts of the sentence most relevant to each aspect [68].

2. Position-aware attention: This mechanism incorporates positional information into the attention weights, giving more importance to words that are closer to the target aspect [72].

3. Self-attention: Self-attention, as used in the transformer architecture, allows the model to attend to different parts of the sentence based on their relevance to each other, capturing long-range dependencies and complex relationships [64].

4. Hierarchical attention: This approach involves applying attention mechanisms at multiple levels, such as word-level and sentence-level, to capture the hierarchical structure of text and its impact on aspect-level sentiment [73].

The choice of attention mechanism depends on the specific requirements of the ABSA task and the characteristics of the dataset. However, it is important to note that attention mechanisms are not a panacea and may introduce challenges. One concern is the potential for attention to be misled by irrelevant or noisy parts of the text, leading to incorrect sentiment predictions [74]. Additionally, the interpretability of attention weights has been questioned, with some studies suggesting that they may not always provide a faithful explanation of the model's decision-making process [75].

Despite these challenges, attention mechanisms remain a powerful tool in the ABSA toolkit, enabling models to effectively capture the nuanced relationships between aspects and sentiments in text. As research in this area continues to evolve, we can expect to see further innovations and improvements in the application of attention mechanisms to ABSA tasks.

2.2.4 Contrastive Learning Techniques

Contrastive learning is a technique used to learn representations by contrasting positive and negative pairs of examples. It has gained prominence in unsupervised and self-supervised learning, where labelled data is scarce or unavailable [76, 77]. The core idea is to bring representations of similar (positive) examples closer together while pushing representations of dissimilar (negative) examples apart.

In the context of ABSA, contrastive learning can be employed to create aspect-specific groups. For instance, pairs of sentences that share the same aspect and sentiment can be considered positive pairs, while those with different aspects or sentiments can be negative pairs. This training strategy encourages the model to learn more discriminative representations, improving its ability to distinguish between different aspects and sentiments [78, 79].

Contrastive learning techniques, such as SimCLR (Simple Framework for Contrastive Learning of Visual Representations) and MoCo (Momentum Contrast), have been adapted to NLP tasks, demonstrating significant improvements in representation learning and downstream performance [80, 81].

The application of contrastive learning to ABSA offers several advantages. Firstly, by learning to distinguish between similar and dissimilar pairs of examples, contrastive learning helps the model to develop more robust and discriminative representations of aspects and sentiments [78]. This is particularly important in ABSA, where the model must be able to accurately identify the sentiment expressed towards specific aspects, even when they are mentioned in complex or ambiguous contexts.

Secondly, contrastive learning can help to alleviate the need for large amounts of labelled data, which is often a bottleneck in ABSA tasks [82]. By leveraging unsupervised or self-supervised techniques, contrastive learning allows the model to learn meaningful representations from unlabeled data, which can then be fine-tuned on a smaller labelled dataset for the specific ABSA task.

Furthermore, contrastive learning can be combined with other techniques, such as data augmentation and transfer learning, to further improve the performance of ABSA models [83, 84]. For example, data augmentation techniques can be used to generate additional positive and negative pairs for contrastive learning, while transfer learning can help to adapt pre-trained models to specific domains or languages.

However, the success of contrastive learning in ABSA depends on several factors, such as the choice of positive and negative pairs, the design of the contrastive loss function, and the quality of the underlying representations [82]. Inappropriate choices in any of these areas can lead to sub-optimal performance or even degraded results.

Additionally, contrastive learning methods can be computationally expensive, particularly when dealing with large datasets or high-dimensional representations [76]. This can limit their applicability in resource-constrained environments or real-time applications.

Despite these challenges, contrastive learning has shown significant promise in improving the performance of ABSA models, particularly in scenarios where labelled data is limited or the target domain is different from the source domain. The MASCoT framework incorporates contrastive learning techniques to generate high-quality, aspect-specific representations that enhance the model's ability to accurately predict sentiment at the aspect level.

As research in this area continues to progress, we can expect to see further innovations in the application of contrastive learning to ABSA, such as the development of more advanced loss functions, the incorporation of domain-specific knowledge, and the integration with other state-of-the-art techniques in NLP.

2.2.5 Temporal Analysis Methods

Temporal analysis in NLP involves understanding how language and sentiments change over time. This is particularly important for tasks such as social media monitoring, where user opinions can shift rapidly in response to events [85, 86].

Temporal dynamics can be captured using various methods, such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and transformers. These models are designed to handle sequential data and can learn dependencies across different time steps [87, 88].

In ABSA, incorporating temporal analysis allows models to track how sentiments associated with different aspects evolve within a sentence or across multiple sentences. This is crucial for applications like real-time sentiment analysis, where understanding the temporal flow of opinions can provide deeper insights into user behaviour and preferences [89, 90].

One approach to temporal analysis in ABSA is to use time-aware models that explicitly incorporate temporal information into the sentiment prediction process. For example, Zhang et al. [89] proposed a time-aware LSTM model that captures the temporal dynamics of aspect-level sentiment by incorporating time-decay attention mechanisms. This allows the model to give more importance to recent sentiment expressions while still considering the overall context of the text.

Another approach is to use temporal convolutional networks (TCNs), which are effective for modelling sequential data in various NLP tasks [91]. TCNs can capture long-range dependencies and temporal patterns in text, making them well-suited for ABSA tasks that involve tracking sentiment over time.

However, temporal analysis in ABSA also presents several challenges. One challenge is the sparsity of temporal data, particularly in domains where sentiment expressions are infrequent or unevenly distributed over time [89]. This can make it difficult for models to learn robust temporal patterns and may require the use of techniques like data augmentation or transfer learning to overcome data limitations.

Another challenge is the complexity of temporal dynamics in sentiment expression. Sentiments can change rapidly and unpredictably in response to events, and the same aspect may be associated with different sentiments at different points in time [86]. Capturing these complex dynamics requires models that are sensitive to the temporal context and can adapt to changing sentiment patterns over time.

Despite these challenges, temporal analysis remains an important area of research in ABSA, particularly as the volume and velocity of opinionated data continue to grow.

CHAPTER 3

Literature Review

This review aims to analyse existing literature and seek to identify gaps and opportunities for further advancement in the field. This, in turn, will inform the development and implementation of our proposed MASCoT (Multi-Aspect Sentiment Analysis with Contrastive Learning and Temporal insights) framework.

In response to the evolving landscape of sentiment analysis, various approaches have been explored to tackle its unique challenges. Hybrid models have gained substantial traction in Aspect-Based Sentiment Analysis (ABSA), offering a promising solution to overcome the limitations of lexicon-based and pure machine learning (ML) approaches. These hybrid models amalgamate the strengths of both methods to deliver a more robust sentiment analysis, as demonstrated by Zhang et al. (2020) [92].

3.1 Review of Existing Literature

This section will provide a comprehensive overview of the advantages and disadvantages of each method, along with their respective references. The table 3.1 shows the extensive compilation and identification of important literature works, and serves as a valuable resource for researchers and practitioners in sentiment Analysis. It covers the progress made in the field, from traditional lexicon-based approaches to more advanced deep learning techniques, while also acknowledging the limitations and challenges that remain to be addressed in future research.

Table 3.1: Comparison of sentiment analysis methods and models

Method/Model	Advantages	Disadvantages	Reference
CNN + Matrix factorization + graph attention network + MLP	Combines lexicon-based and machine-learning strengths for robust sentiment analysis. The graph attention network captures intricate relationships between aspects and sentiment words.	May rely heavily on training data, which can be challenging to obtain for specific domains. The model's adaptability to new domains may be limited.	[93]
Opinion Finder (MPQA)	Recognizes subjectivity and sentiment nuances by leveraging a comprehensive subjectivity lexicon and extraction patterns. Provides fine-grained sentiment analysis.	Heavy reliance on training data, which may not be readily available for all domains. The model's adaptability to new domains may be limited.	[94]
SentiWordNet	Offers a systematic approach to lexicon-based sentiment analysis by assigning scores to WordNet synsets. Provides wide coverage of English words and phrases.	Limited by WordNet's language coverage, which may not include domain-specific terms. Scores may not always reflect context-dependent sentiment.	[95]
Stanford Recursive Deep Model (RNTN)	Excels in capturing the compositional structure of complex sentences using a recursive neural network. Outperforms traditional bag-of-words models.	Demands substantial computational resources and time due to its recursive architecture. Model interpretability may be limited compared to lexicon-based approaches.	[96]

Continued on next page

Table 3.1 – continued from previous page

Method/Model	Advantages	Disadvantages	Reference
Multilingual BERT (mBERT) & XLM-RoBERTa (XLM-R)	Enables multilingual sentiment analysis by leveraging pre-trained language models. Achieves state-of-the-art performance on various benchmarks.	Performance may vary depending on hyperparameters and pre-training data quality. Fine-tuning for specific languages and domains may be necessary.	[97]
Character-based DBLSTM	Achieves high accuracy by capturing character-level features. Robust to out-of-vocabulary words and misspellings.	Requires preprocessing of special characters and numbers. May struggle with long-range dependencies compared to word-level models.	[98]
1D-CNN & Recurrent Networks	Achieves high accuracy by combining 1D-CNN for local feature extraction and GRU for sequence modeling. Outperforms individual CNN and RNN models.	Architecture can be complex with multiple layers and hyperparameters. Model interpretability may be limited compared to simpler models.	[99]
SAB-LSTM	Outperforms conventional LSTMs by incorporating self-attention and bidirectional mechanisms. Captures both local and global context.	Requires additional layers for optimization, increasing model complexity and training time. Interpretability may be limited.	[100]
Support Vector Machine, Multinomial Naïve Bayes, LSTM, BERT	BERT achieves high accuracy using pre-trained language representations. Outperforms traditional ML models like SVM and Naïve Bayes.	Different models may require varying preprocessing steps. Fine-tuning BERT for specific domains may be necessary for optimal performance.	[101]

Continued on next page

Table 3.1 – continued from previous page

Method/Model	Advantages	Disadvantages	Reference
LSTM for Bangla Tweets	Achieves high accuracy in sentiment analysis of Bangla tweets by capturing the sequential nature of text data. Outperforms traditional ML models.	Data cleaning and preprocessing can be challenging for low-resource languages. Model architecture may require optimization for better performance.	[102]
BERT Large with UDA	Achieves high classification accuracy and demonstrates effectiveness in semi-supervised learning. Provides robust performance with large datasets.	Requires significant computational resources and extensive fine-tuning. Can be complex to implement and train.	[103, 104]
Graph Neural Network, SenticNet	Preserves contextual representation by modeling word relationships using a graph structure. SenticNet provides a rich knowledge base.	Potential scalability issues with large graphs and complex architectures. SenticNet has limitations in domain-specific adaptability.	[105]
IAN	Interactively learns target and context representations, identifying important words in the context and targets.	Relies on word embedding quality and may struggle with long-range dependencies.	[106]
TNet	Employs a CNN layer to extract salient features from transformed word embeddings, capturing context-sensitive information.	May not effectively capture intricate relationships between aspects and opinion terms.	[107]

Continued on next page

Table 3.1 – continued from previous page

Method/Model	Advantages	Disadvantages	Reference
RAM	Utilizes multiple attention mechanisms to capture sentiment features separated by a long distance, making it more robust against irrelevant information.	Model complexity may increase with the number of attention mechanisms, affecting training time and interpretability.	[108]
MGAN	Leverages a fine-grained attention mechanism to capture word-level interactions between aspects and context words.	Relies heavily on word embedding quality and may not effectively capture global context.	[109]
DGEDT	Dynamically models sentiment dependencies between the target and its context by constructing a directed graph.	Performance may be sensitive to the quality of dependency parsing results.	[110]
TD-LSTM	Incorporates target information into representation learning, capturing target-specific context for ABSA. Outperforms previous methods on SemEval-2014.	Performance may be limited when dealing with complex sentence structures and long-range dependencies.	[111]
ATAE-LSTM	Combines attention with LSTM to focus on important words in the context of a given aspect. Improves the model's ability to capture relevant sentiment information.	The attention mechanism may not fully capture complex relationships between aspects and sentiment expressions.	[68]

Continued on next page

Table 3.1 – continued from previous page

Method/Model	Advantages	Disadvantages	Reference
GCNN	Incorporates gate mechanisms to selectively output sentiment features based on the given aspect. Effectively handles implicit aspects and sentiments.	Performance may be sensitive to the quality of aspect-level annotations used for training.	[112]
TABSA	Integrates commonsense knowledge from ConceptNet to enhance representation learning. Achieves state-of-the-art results on SemEval-2014 and Twitter datasets.	Incorporating external knowledge may introduce noise and increase model complexity. Performance may be limited by the coverage and quality of the commonsense knowledge base.	[113]
Multi-task BERT	Jointly learns aspect term extraction and sentiment classification, leveraging pre-trained language models. Outperforms previous approaches on SemEval-2014.	Fine-tuning BERT for specific ABSA tasks may be computationally expensive. Model interpretability may be limited compared to models with explicit aspect and sentiment representations.	[114]
Aspect-Pair Supervised Contrastive Learning (APSCL)	Captures latent relationships between multiple aspects, enhancing aspect representation through relationship optimization. Improves accuracy and F1 score significantly.	Primarily focuses on relational representations. May not fully address handling diverse and complex sentiment interactions across different domains.	[115]

Continued on next page

Table 3.1 – continued from previous page

Method/Model	Advantages	Disadvantages	Reference
Contrastive Learning for ABSA	Enhances aspect-specific sentiment discrimination. Sentiment-based and augmentation-based methods show improvements in public datasets.	Effectiveness varies with data augmentation strategies. Combining both methods does not always yield better performance. Generalizability across different datasets needs validation.	[6]

3.2 Existing Knowledge Gaps

Despite the substantial advancements in aspect-based sentiment analysis (ABSA), several critical knowledge gaps persist in the field. These gaps highlight the limitations of current methodologies and provide a rationale for the development of more advanced models like the MASCoT framework.

Capturing Long-Range Dependencies and Complex Sentence Structures

One notable contribution to the field is the work by Tang et al. [111], who proposed the Target-Dependent LSTM (TD-LSTM) model for ABSA. The TD-LSTM incorporates target information into the representation learning process, allowing the model to capture target-specific context when determining the sentiment towards a particular aspect. By considering the relationship between the target and its surrounding context, the TD-LSTM outperformed previous methods on the SemEval-2014 dataset. However, the model's performance may be limited when dealing with complex sentence structures and long-range dependencies. This limitation underscores a significant gap in the ability of current models to effectively handle intricate sentence constructions and extended dependencies, which are common in natural language.

Effectiveness of Attention Mechanisms

Attention mechanisms have emerged as a powerful technique for improving the performance of sentiment analysis models. Wang et al. [68] introduced the Attention-based LSTM with Aspect Embedding (ATAE-LSTM), which combines an attention mechanism with LSTM to focus on important words in the context of a given aspect. The attention mechanism enables the model to assign higher weights to words that are more relevant to the aspect, thereby improving its ability to capture relevant sentiment information. While the ATAE-LSTM has shown promising results, the attention mechanism may not fully capture the complex relationships between aspects and sentiment expressions. This indicates a gap in the current understanding and application of attention mechanisms, particularly in capturing nuanced interactions within sentences.

External Knowledge Integration

Another significant contribution is the Gated Convolutional Neural Network (GCNN) proposed by Xue et al. [112]. The GCNN incorporates gate mechanisms to selectively output sentiment features based on the given aspect, effectively handling implicit aspects and sentiments. By learning to attend to relevant sentiment signals and suppress irrelevant ones, the GCNN achieves improved performance in ABSA tasks. However, the model's performance may be sensitive to the quality of the aspect-level annotations used for training. Similarly, Ma et al. [113] proposed the Targeted Aspect-Based Sentiment Analysis (TABSA) model, which incorporates commonsense knowledge from ConceptNet to enrich the representation learning process. While effective, the incorporation of external knowledge may introduce noise and increase the model's complexity, and its performance may be limited by the coverage and quality of the commonsense knowledge base. These points highlight gaps related to the dependency on high-quality annotations and the challenges of integrating external knowledge effectively.

Computational Efficiency of Pre-Trained Language Models

Pre-trained language models, such as BERT [116], have revolutionized the field of natural language processing, including sentiment analysis. Li et al. [117] proposed a multi-task BERT model that jointly learns aspect term extraction and sentiment classification. By fine-tuning BERT for specific ABSA tasks, the model outperforms previous approaches on the SemEval-2014 dataset, demonstrating the effectiveness of leveraging pre-trained language representations. However, fine-tuning BERT for ABSA tasks can be computationally expensive, and the model's interpretability may be limited compared to models with explicit aspect and sentiment representations. This gap underscores the need for models that balance performance with computational efficiency and interpretability.

Handling Multiple Aspects with Conflicting Sentiments

The incorporation of temporal dynamics to capture the evolution of sentiments within a text remains an underexplored area in ABSA. Most existing approaches focus on sentiment classification at the sentence or aspect level, ignoring the temporal flow of opinions. Models like the Hierarchical Attention Network (HAN) [118] and the Multi-grained Attention Network (MGAN) [109] have made efforts to capture the hierarchical structure of documents and the interactions between aspects and context words. However, they do not explicitly model the temporal dynamics of sentiments.

Additionally, the ability to handle multiple aspects with potentially conflicting sentiments within a single text remains a challenge in ABSA. Models like the Aspect Alignment Network (AAN) [119] and the Multi-Task Learning Network (MTLN) [120] have attempted to address this issue by aligning aspects and sentences or learning shared representations across different tasks. However, there is still room for improvement in effectively capturing the complex interactions between multiple aspects and their associated sentiments. These gaps highlight the necessity for models that can dynamically adjust to the temporal and multi-faceted nature of real-world data.

Limitations of Contrastive Learning Techniques

Contrastive learning has been explored in ABSA to improve the discrimination of aspect-specific sentiments. Xu and Wang [6] demonstrated that both sentiment-based and augmentation-based contrastive learning lead to consistent improvements in ABSA tasks. However, the augmentation-based approach's effectiveness heavily depends on the data augmentation strategies employed, and combining both approaches does not always result in better performance. Additionally, the generalizability of these techniques across different datasets and modalities needs further validation. Similarly, Li et al. [115] proposed an Aspect-Pair Supervised Contrastive Learning (APSCL) model to capture relationships between multiple aspects in the sentiment subspace. While APSCL shows improvements, it primarily focuses on relational representations and may not fully address the need for handling diverse and complex sentiment interactions across different domains. These studies indicate that while contrastive learning offers promise, there are gaps in its application, particularly in terms of generalizability and integration with other advanced NLP techniques.

In conclusion, while substantial progress has been made in ABSA, significant knowledge gaps remain, particularly in capturing long-range dependencies, leveraging attention mechanisms effectively, integrating external knowledge, improving computational efficiency, and handling temporal dynamics and conflicting sentiments. Addressing these gaps is crucial for developing more accurate, interpretable, and efficient sentiment analysis models. The MASCoT model aims to address these challenges, offering a comprehensive solution to the existing limitations in ABSA.

3.3 Proposed Innovations

The current research introduces the Multi-Aspect Sentiment Contrastive Learning (MASCoT) framework, aimed at addressing several key limitations identified in existing ABSA methodologies. MASCoT is designed to enhance the robustness, interpretability, and temporal awareness of multi-aspect sentiment analysis models. This section outlines how MASCoT tackles these challenges through innovative techniques and methodologies, advancing the current landscape inspired by previous works.

Improving Model Interpretability

Pre-trained language models like BERT have significantly advanced sentiment analysis but often lack interpretability, making it challenging to understand the underlying reasons for their predictions. MASCoT addresses this by incorporating a novel attention mechanism specifically designed to highlight the most relevant words and aspects contributing to sentiment classification. Unlike previous models such as the Attention-based LSTM with Aspect Embedding (ATAE-LSTM) by Wang et al.[68], which may not fully capture complex relationships between aspects and sentiment expressions, MASCoT's attention mechanism ensures more precise results. This mechanism allows for aspect-relative embeddings, where aspect embeddings are distributed across the sentence to enhance the weight of aspect and its related terms.

3.3.1 Temporal Analysis

MASCoT incorporates temporal analysis methods to capture the evolution of sentiments within a text, which is particularly useful for real-time sentiment analysis applications. By employing a temporal overlapping window segmentation approach, MASCoT maintains context around aspect terms even in long and complex sentences. This method allows the model to capture the progression of sentiments and topics throughout the sentence, crucial for accurate sentiment analysis. While models like the Hierarchical Attention Network (HAN) by Yang et al. and the Multi-grained Attention Network (MGAN) by Fan et al. have aimed to capture the hierarchical structure of documents and interactions between aspects and context words, they do not explicitly model temporal dynamics. MASCoT addresses this gap by allowing the model to understand how sentiments evolve, enhancing its applicability in domains such as social media monitoring and real-time feedback analysis.

3.3.2 Handling Multiple Aspects with Conflicting Sentiments

Models like the Aspect Alignment Network (AAN) by Yang et al. and the Multi-Task Learning Network (MTLN) by Li et al. have attempted to align aspects and sentences or learn shared representations across different tasks. However, there is still room for improvement in capturing the complex interactions between multiple aspects and their associated sentiments. MASCoT addresses this by using a multi-head attention mechanism that independently focuses on each aspect, ensuring accurate sentiment capture even when multiple aspects are present in a single sentence.

3.3.3 Enhancing Aspect-Specific Sentiment Discrimination

To enhance aspect-specific sentiment discrimination, MASCoT leverages a novel attention mechanism framework inspired by the works of Li et al. and Xu and Wang. These studies demonstrated the effectiveness of contrastive learning for ABSA; however, MASCoT extends this by integrating innovative attention mechanisms and pair generation techniques. By creating positive and negative pairs based on aspect and sentiment labels, the model effectively learns to differentiate between similar and dissimilar sentiment contexts. This approach surpasses the limitations highlighted in APSCL, which primarily focuses on relational representations without adequately addressing complex sentiment interactions across various domains. MASCoT incorporates dynamic modeling techniques to capture intricate relationships between multiple aspects and their associated sentiments, ensuring better performance in complex, real-world scenarios.

3.3.4 Generalizability across Domains

Studies indicate that while contrastive learning offers promise, there are gaps in its application, particularly in terms of generalizability and integration with other advanced NLP techniques. MASCoT addresses this by employing robust data augmentation techniques such as back translation and synonym replacement. This increases the lexical diversity of the training data and improves the model's robustness. Introducing variability in the training data ensures that MASCoT can handle different expressions and variations of the same aspects, leading to better performance in real-world scenarios.

In conclusion, while existing models have made significant strides in ABSA, challenges such as adaptability to new domains, interpretability, incorporation of temporal dynamics, and handling multiple aspects with conflicting sentiments remain. The MASCoT framework builds upon the existing body of knowledge by addressing these challenges and paving the way for more accurate and comprehensive sentiment analysis systems.

CHAPTER 4

Methodology

In this chapter, we presented our multi-aspect Aspect-Based Sentiment Analysis (ABSA) framework called MASCoT (Multi-Aspect Sentiment Analysis with Contrastive Learning and Temporal Insights). The overview of our framework's architecture was demonstrated in Figure 4.1. It could be observed that the ABSA framework, in its entirety, could be divided into several components with different functionalities.

Firstly, the input data, consisting of sentences and aspect terms, underwent pre-processing and augmentation techniques to standardize sentiment labels, apply data augmentation, handle multiple aspects, and split the data into training, validation, and test sets [121, 122].

Next, the pre-processed data was passed through the embedding generation and attention mechanism module. This module utilized RoBERTa tokenization and precomputed embeddings to generate contextualized representations of sentences and aspect terms [57]. A novel multi-head attention module was employed to incorporate aspect information into the embedding generation process, enabling the model to capture aspect-specific sentiment nuances [64]. Additionally, a temporal overlapping window segmentation approach was applied to capture changing sentiments over time [123].

The generated windows were then fed into the contrastive learning and model architecture module. This module included a sentiment-based pair generation component that created positive and negative sentence pairs based on their focus aspect and sentiment labels [80]. The contrastive model architecture consisted of a BERT-based encoder, fully connected layers, and aspect-specific sentiment prediction [56]. The contrastive learning approach encouraged the model to learn discriminative sentiment representations by contrasting positive and negative sentence pairs [124].

Finally, the training and evaluation methodology was employed to optimize the model's performance. The model was trained using the focal loss function to handle class imbalance [125], and validation techniques such as early stopping and hyperparameter tuning were applied. The trained model was evaluated using various metrics, including accuracy, precision, recall, F1 score, and confusion matrix, to assess its performance on the test set. Visualizations of the training and validation results were provided for analysis and interpretation.

In the following sections, we introduced each component of the MASCoT framework in detail.

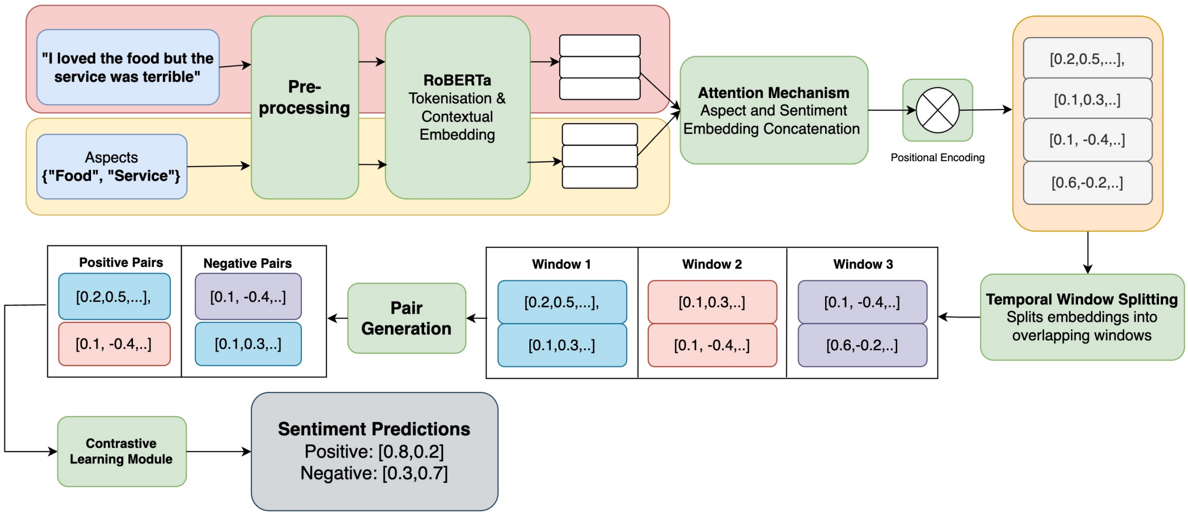

FIGURE 4.1: Overall architecture of the MASCoT framework.

4.1 Input Processing Module

In our multi-aspect ABSA framework, the pre-processing step played a crucial role in preparing the input data for effective model training and evaluation [126]. The primary objective of this step was to transform the raw input data into a structured and standardized representation that could be easily consumed by the subsequent components of the framework.

4.1.1 Data Preprocessing

In our data preprocessing pipeline, we made a conscious decision to retain stop words, which are commonly removed in many NLP tasks. This decision was based on the observation that stop words can carry valuable sentiment information and contribute to the overall sentiment polarity of a text. Several studies have highlighted the importance of stop words in sentiment analysis tasks [127, 128]. By retaining stop words, we aimed to preserve the nuances and contextual cues present in the text, enabling our models to capture the sentiment more accurately. Moreover, the decision to retain stop words aligned with the objective of the MAMS dataset, which focuses on multi-aspect sentences with potentially complex sentiment expressions [129]. In such scenarios, stop words can play a crucial role in disambiguating the sentiment associated with different aspects within a single sentence. By incorporating data augmentation techniques and retaining stop words during preprocessing, we aimed to create a robust and representative training dataset that captures the intricacies of sentiment expression, ultimately enhancing the performance of our aspect-based sentiment analysis models.

The pre-processing step began by parsing the XML data files, which contained sentences annotated with aspect terms and their corresponding sentiment labels. This parsing process involved extracting the relevant information and organizing it into a structured format, such as a pandas DataFrame. By doing so, we created a unified representation of the input data that was consistent across different datasets and facilitated efficient data manipulation and analysis.

One of the key challenges in sentiment analysis is the inconsistency in sentiment label representations across different datasets. To address this issue, we introduced a sentiment label standardization process. This process mapped the original sentiment labels to a common numeric representation, such as 0 for "negative" and 1 for "positive". By standardizing the sentiment labels, we ensured that the model could learn from a consistent and meaningful representation of sentiment, regardless of the dataset's original labelling scheme.

4.1.2 Data Augmentation Techniques

Another important aspect of our pre-processing methodology was data augmentation. In real-world scenarios, the available training data for ABSA tasks may be limited, leading to potential overfitting and poor generalization of the model. To mitigate this issue, we employed two data augmentation techniques: back translation and synonym replacement.

4.1.2.1 Back-Translation

One of the augmentation methods we utilized was back-translation, which involved translating the original text to an intermediate language and then translating it back to the original language [130, 131, 132].Fig 4.2 illustrates the procedure employed.

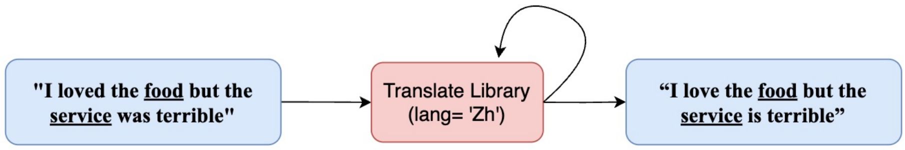

FIGURE 4.2: Back Translation with 'Chinese' as Intermediary

This technique has been shown to introduce valuable linguistic variations while preserving the overall meaning and sentiment of the text. By applying back-translation, we aimed to generate semantically similar but syntactically diverse training examples, improving the model's ability to generalize across different language patterns.

4.1.2.2 Synonym Replacement

Another augmentation strategy we employed was synonym replacement, where we substituted certain words in the original text with their synonyms [133, 134, 132]. This approach helped to increase the lexical diversity of the training data, exposing the model to different word choices that express similar meanings and sentiments. We utilized pre-trained language models, such as BERT [56], to identify suitable synonyms while considering the context of the sentence, ensuring that the substitutions preserved the original sentiment. Fig 4.3 illustrates the procedure employed.

FIGURE 4.3: Synonym Replacement

The effectiveness of data augmentation techniques, including back-translation and synonym replacement, in improving sentiment analysis performance has been extensively studied and validated in various research works [135, 136, 137, 122]. These techniques have been shown to enhance the generalization ability of sentiment analysis models, particularly when dealing with limited or imbalanced training data.

Finally, to ensure proper evaluation and tuning of the model, we split the preprocessed data into training, validation, and test sets. This split allowed us to assess the model's performance on unseen data and make informed decisions regarding hyperparameter tuning and model selection.

By incorporating these novel preprocessing techniques, our multi-aspect ABSA framework addressed the challenges of sentiment label inconsistency, limited training data, and multiple aspects per sentence.

The standardization of sentiment labels, data augmentation through back translation and synonym replacement, and aspect-level granularity enabled the framework to learn robust and generalizable sentiment representations. These preprocessing steps laid the foundation for the subsequent components of the framework, where the preprocessed data was transformed into embeddings and used to train the ABSA model effectively.

4.2 Contextual Embeddings Retrieval Module

In this section, we delved into the details of the Contextual Embeddings Retrieval Module, which served as a crucial component in our aspect-based sentiment analysis pipeline. This module was responsible for preprocessing the input data, generating contextualized embeddings, and preparing the necessary inputs for the subsequent attention mechanism. This section was divided into two sub-sections: RoBERTa Embedding Generation and Attention Mechanism Architecture. Fig 4.4

4.2.1 RoBERTa Embedding Generation

The first step in the Contextual Embeddings Retrieval Module was the contextualization of the input sentences and aspects using the RoBERTa (Robustly Optimized BERT Pretraining Approach) model [57]. RoBERTa is a state-of-the-art language model that has shown remarkable performance in various natural language processing tasks, including sentiment analysis [56].

The choice of using RoBERTa in our experiment was motivated by its superior performance, as shown in several studies. Compared to other pre-trained language models, such as BERT [56], Electra [138], XLNet [139], and DistilBERT [140], RoBERTa has empirically demonstrated effectiveness in various natural language processing tasks, owing to its larger batch sizes, dynamic masking, and longer training time compared to BERT.

We employed the RoBERTa tokenizer to tokenize the input sentences and aspects into sub-word units. The function took a sentence, an aspect, the tokenizer, and the maximum sentence length as input. First, the tokenizer split the text into individual tokens and applied byte-pair encoding (BPE) [141] to handle out-of-vocabulary words and generate sub-word units.

After tokenization, custom preprocessing steps were applied to handle specific characters, such as removing the "Ġ" character from the beginning of tokens and replacing "ł" and "Â" with "t" and "a",

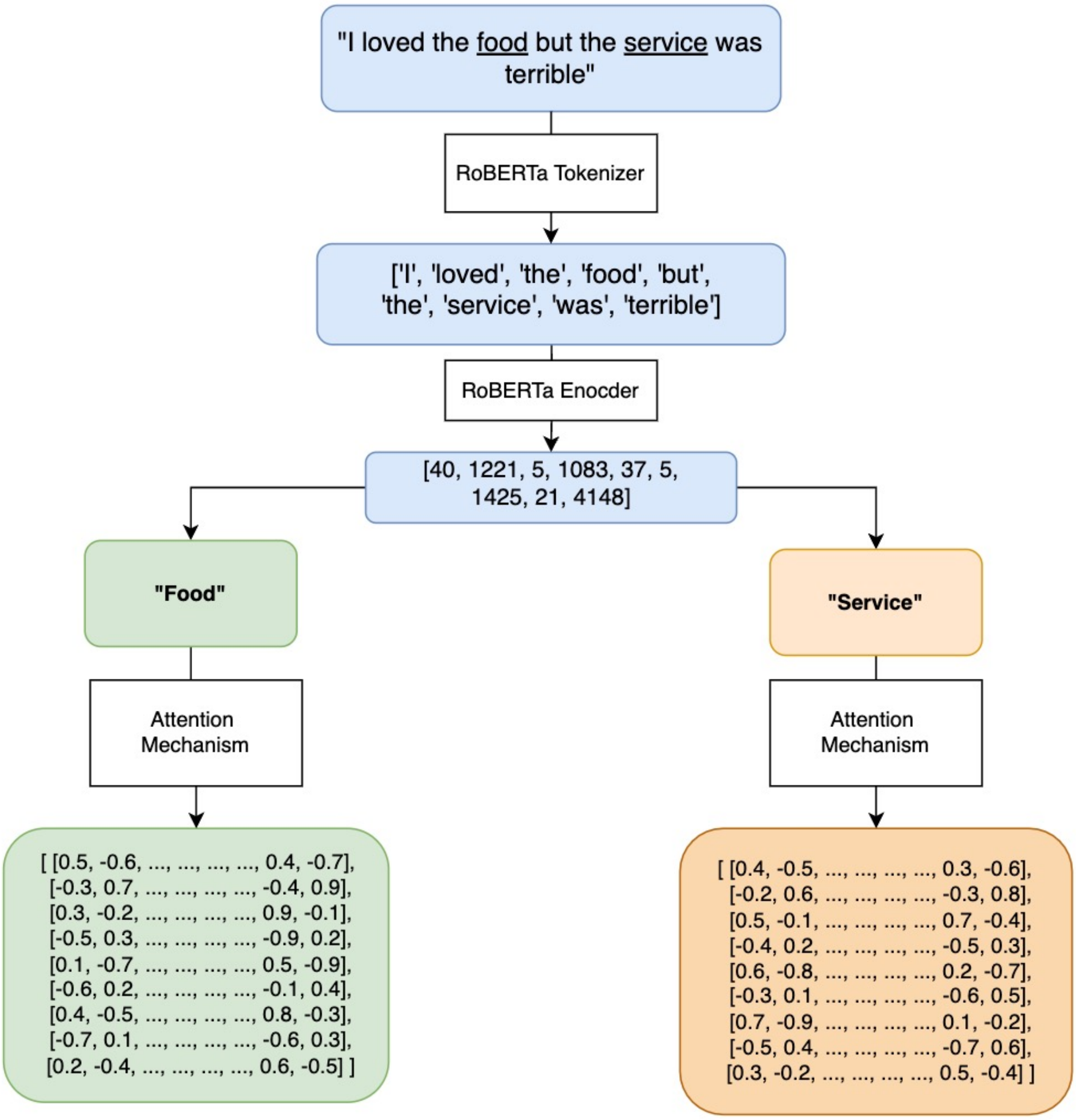

FIGURE 4.4: Overview of Embedding Module

respectively. These preprocessing steps helped to normalize the text and ensure consistency across the dataset.

To prepare the input for the RoBERTa embedding model, the function constructed the input sequence by concatenating the sentence tokens, aspect tokens, and special tokens (e.g., '[CLS]' and '[SEP]') in a specific format. The input sequence was represented as follows:

$$X = [\text{CLS}] \oplus S \oplus [\text{SEP}] \oplus A \oplus [\text{SEP}]$$

where X is the input sequence, S represents the sentence tokens, A represents the aspect tokens, and $\oplus$ denotes the concatenation operation. The '[CLS]' token was added at the beginning of the sequence, and '[SEP]' tokens were used to separate the sentence and aspect tokens.

An aspect mask, denoted as M_A, was created to indicate the positions of the aspect tokens in the input sequence. The aspect mask is a binary vector where $M_A[i] = 1$ if the i-th token belongs to the aspect, and $M_A[i] = 0$ otherwise.

Finally, the function converted the tokens to their corresponding IDs using the tokenizer and created an attention mask, denoted as M_X, where $M_X[i] = 1$ if the i-th token is a valid token, and $M_X[i] = 0$ if it is a padding token. The attention mask was used to indicate the valid positions in the input sequence during the attention computation and helped to focus on the relevant aspect-related information.

The resulting input sequence, aspect mask, and attention mask were then padded to the maximum sequence length to ensure uniform input size across the dataset. The padding operation can be represented as follows:

$$X_{\text{padded}} = [X_1, X_2, \ldots, X_n, \text{PAD}, \ldots, \text{PAD}]$$

$$M_{A,\text{padded}} = [M_{A,1}, M_{A,2}, \ldots, M_{A,n}, 0, \ldots, 0]$$

$$M_{X,\text{padded}} = [M_{X,1}, M_{X,2}, \ldots, M_{X,n}, 0, \ldots, 0]$$

where PAD represented the padding token, and n was the length of the input sequence before padding.

The RoBERTa model took the input sequence X_{padded}, attention mask $M_{X,\text{padded}}$, and token type IDs (if applicable) as input and produced a sequence of hidden states H. Each hidden state h_i corresponded to the contextualized representation of the i-th token in the input sequence. The hidden states can be represented as follows:

$$H = [h_{\text{CLS}}, h_1, h_2, \ldots, h_n, h_{\text{PAD}}, \ldots, h_{\text{PAD}}]$$

where h_{CLS} is the hidden state corresponding to the '[CLS]' token, h_i is the hidden state of the i-th token, and h_{PAD} represents the hidden states of the padding tokens.

For each sample in the batch, the function extracted the token embeddings from the last hidden state of the RoBERTa model. To obtain a single sentence embedding vector, two common pooling strategies were used: average pooling and max pooling. Average pooling involved computing the mean of the token embeddings, while max pooling involved selecting the maximum value for each dimension across the token embeddings. The pooling operation can be represented as follows:

$$e_S = \frac{1}{n}\sum_{i=1}^{n} h_i \quad \text{(average pooling)}$$

$$e_S = \max_{i=1}^{n} h_i \quad \text{(max pooling)}$$

where e_S is the sentence embedding vector, and n is the length of the input sequence.

Additionally, the function computed the aspect embedding by multiplying the token embeddings with the aspect mask and taking the sum, normalized by the number of aspect tokens. The aspect embedding can be represented as follows:

$$e_A = \frac{\sum_{i=1}^{n} h_i \cdot M_{A,i}}{\sum_{i=1}^{n} M_{A,i}}$$

where e_A is the aspect embedding vector, and $M_{A,i}$ is the aspect mask value for the i-th token.

The embedding generation step transformed the tokenized and pre-processed input data into dense vector representations that captured the semantic information of the sentences and aspects. These embeddings served as the foundation for the subsequent attention mechanism, which leveraged them to capture the contextual relationships between words and aspects.

4.2.2 Multi-Head Attention Mechanism Architecture

In our model architecture, we incorporated a multi-head self-attention mechanism to capture the contextual relationships between words and aspects within the input sentences. The attention mechanism has become a fundamental component in various natural language processing tasks since its introduction in the Transformer architecture [64]. It enables the model to focus on the most relevant parts of the input when generating output representations, thereby enhancing the model's ability to capture long-range dependencies and contextual information. In essence, some weight of the 'aspect' was added to other words in the sentence, to make make a connection between to keep the embeddings contextual. Figure 4.5 represents the aforementioned idea.

In the context of aspect-based sentiment analysis, the attention mechanism plays a crucial role in identifying the sentiment polarity associated with specific aspects within a sentence [5]. By attending to the relevant context words surrounding an aspect, the model can better understand the sentiment expressed towards that aspect, leading to improved classification performance.

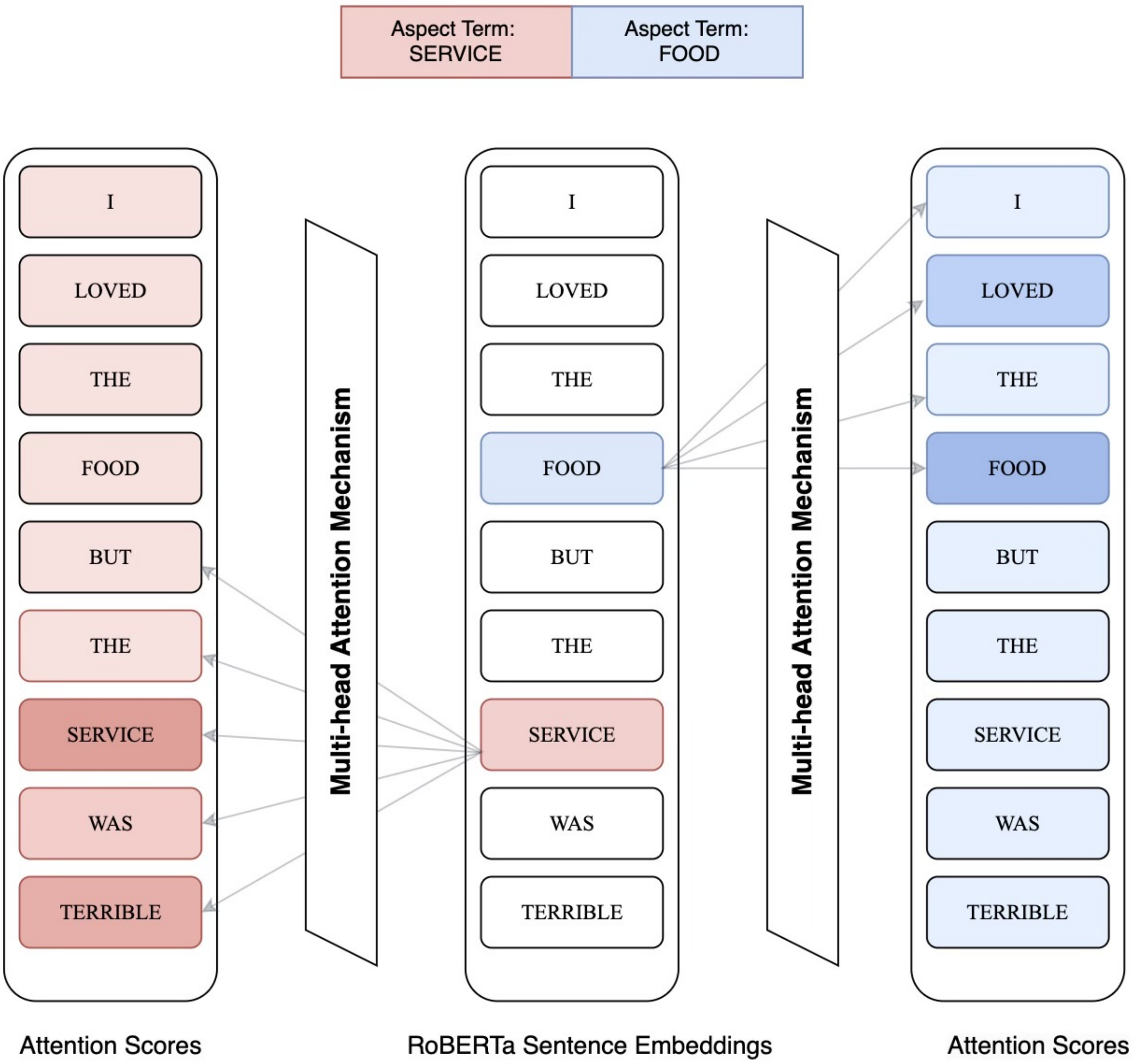

FIGURE 4.5: Attention visualization of RoBERTa embeddings

We employed a custom MultiheadAttentionWithAspect module, which extended the standard multi-head attention mechanism proposed by Vaswani et al. (2017) [64]. The key difference in our approach was the incorporation of aspect embeddings into the query computation. This modification allowed the model to explicitly consider the aspect information when attending to the relevant context words. This approach was particularly beneficial when we split the embeddings into windows, a process that was discussed in a subsequent section. By ensuring that every slice of the window contained its respective aspect information, we guaranteed that even if a sliced window did not contain explicit sentiment or aspect terms, it still captured the contextual relevance to the aspect from which it was derived, i.e., it remained connected to its expressed aspect. This enhanced the model's ability to maintain aspect-specific context and sentiment information throughout the entire sentence, leading to more accurate sentiment classification. Fig 4.6 illustrates the architecture of the network.

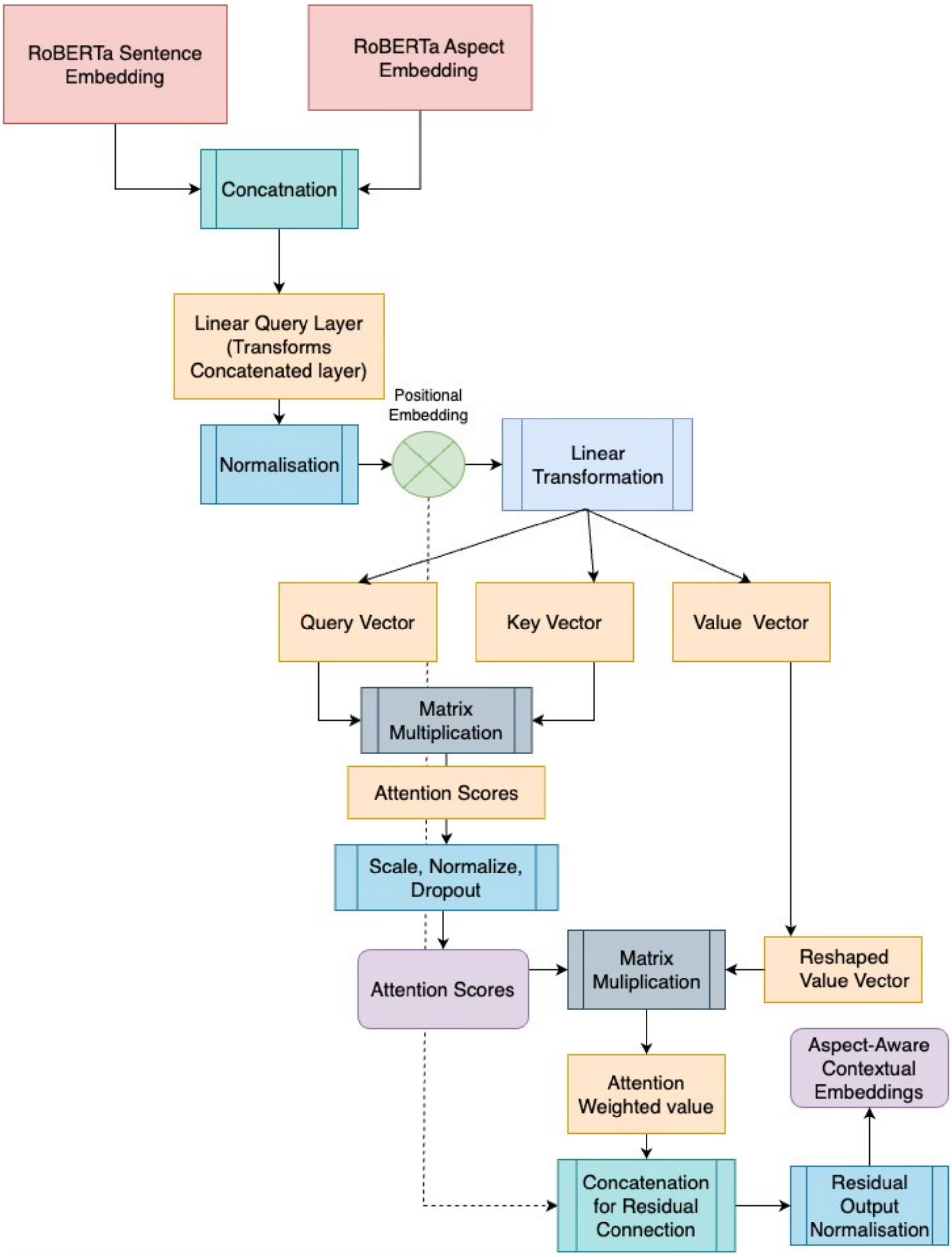

FIGURE 4.6: Attention Mechanism Architecture

The network took as input the sentence embeddings and aspect embeddings. The sentence embeddings, denoted as $X \in \mathbb{R}^{n \times d}$, represented the contextual representations of the words in the input sentence, where n is the sequence length and d is the embedding dimension. The aspect embeddings, denoted as $A \in \mathbb{R}^{m\times d}$, represented the specific aspects for which we aimed to determine the sentiment polarity, where m is the number of aspects.

The module first projected the input embeddings into query (Q), key (K), and value (V) representations using learnable linear transformations:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d\times d}$ are the learnable projection matrices.

The query representation was further modified by concatenating it with the aspect embeddings and passing it through an additional linear transformation:

$$Q' = [Q; A]W^{Q'}$$

where $[;]$ denotes concatenation, and $W^{Q'} \in \mathbb{R}^{2d\times d}$ is a learnable projection matrix. This step ensured that the aspect information was explicitly incorporated into the attention computation. The attention scores were then computed using the scaled dot-product attention [64]:

$$\text{Attention}(Q', K, V) = \text{softmax}\left(\frac{Q'K^T}{\sqrt{d_k}}\right)V$$

where d_k is the dimension of the key vectors, and the scaling factor $\frac{1}{\sqrt{d_k}}$ was used to prevent the dot products from becoming too large. The attention scores indicated the degree of relevance or importance between the words in the input sentence, allowing the model to selectively focus on the most relevant parts of the input when computing the contextual representations.

To capture more complex relationships and attend to different aspects of the input simultaneously, we employed a multi-head attention mechanism [64]. This involved splitting the attention computation into h parallel heads, each attending to different subspaces of the input embeddings:

$$\text{MultiHead}(Q', K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

where

$$\text{head}_i = \text{Attention}(Q'W_i^{Q'}, KW_i^K, VW_i^V),$$

and $W_i^{Q'}, W_i^K, W_i^V \in \mathbb{R}^{d\times d_h}$, and $W^O \in \mathbb{R}^{hd_h\times d}$ are learned projection matrices, with $d_h = \frac{d}{h}$ being the dimensionality of each head.

Furthermore, we incorporated positional encodings [64] into the query representation to inject information about the relative positions of the words in the input sentence. Positional encodings were computed using sine and cosine functions of different frequencies:

$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right)$$
$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right)$$

where pos is the position, i is the dimension, and d_{model} is the embedding dimension. The positional encodings were added to the query representation:

$$Q'' = Q' + \mathrm{PE}$$

This step was important because the self-attention mechanism is inherently permutation-invariant, and positional encodings help the model to distinguish between different word orders and capture sequential dependencies.

The output of the MultiheadAttentionWithAspect module was a contextualized representation of the input sentence, where each word's representation was enriched with information from the relevant context words and the associated aspects. This contextualized representation, denoted as $Z \in \mathbb{R}^{n \times d}$, was obtained by applying a linear transformation followed by layer normalization [142] to the concatenated outputs from the multiple heads:

$$Z = \mathrm{LayerNorm}(\mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O)$$

Layer normalization was applied to normalize the activations and stabilize the training process. Given an input vector x, layer normalization was defined as:

$$\mathrm{LN}(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \times \gamma + \beta$$

where μ and σ^2 were the mean and variance of the input vector, ϵ was a small constant for numerical stability, and γ and β were learned scaling and shifting parameters. The resulting contextualized representation Z was obtained in a dictionary, with the keys being a tuple of the sentence ID and aspect. This allowed for efficient retrieval of the embeddings during the subsequent stages of the model.

The MultiheadAttentionWithAspect module served as a critical foundation for subsequent stages of our model, particularly for the Pair Generation and Contrastive Learning modules. These stages were essential for enhancing the model's ability to differentiate between various sentiment polarities associated with different aspects within the input sentences.

4.3 Contrastive Learning Module

The Contrastive Learning Module was pivotal in our framework, designed to improve the model's ability to distinguish between different sentiment polarities associated with various aspects. This module utilized the pairs generated through the Temporal Window Splitting technique, forming positive pairs from similar sentiment contexts and negative pairs from contrasting sentiment contexts. By leveraging contrastive learning principles, the model learned to bring similar sentiment representations closer together while pushing dissimilar ones apart.

4.3.1 Temporal Window Splitting

The Temporal Window Splitting module was an innovative approach that we adopted to enhance our aspect-based sentiment analysis framework. This technique involved dividing the contextual embeddings of sentences into smaller overlapping windows, allowing for a more granular analysis of sentiment towards specific aspects within sentences. The concept of window splitting was inspired by techniques used in time-series analysis and speech processing, where data is segmented into smaller windows to capture local patterns and trends [143, 144].

By splitting sentences into temporal windows, we aimed to capture the changing sentiment and topic in a more sequential manner. The primary advantage of temporal window splitting was that it allowed the model to maintain context around aspect terms even when the sentences were long and complex. This method provided the model with a sequence of smaller contexts, which helped in understanding how sentiment and topics evolved throughout the sentence. Capturing these temporal dynamics was crucial for accurate sentiment analysis, especially in complex sentences where sentiments could shift from positive to negative (or vice versa) across different parts of the sentence. The procedure was depicted in Figure 4.7.

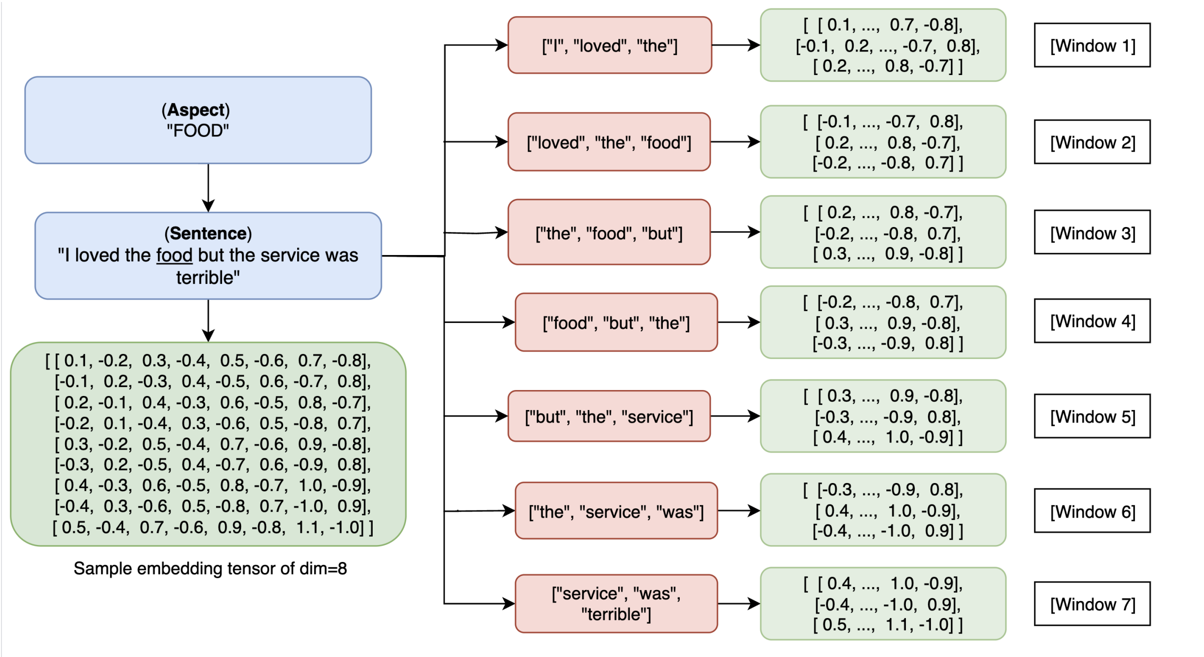

FIGURE 4.7: Depiction of Window-generation of Attention Mechanism Output

4.3.1.1 Window and Stride Sizing

The window size was dynamically adjusted based on the length of the sentence tokens, typically set to one-fourth of the total token length, with a minimum size of 3 tokens in each window. This ensured that each window was large enough to capture meaningful context but small enough to provide granularity. The stride size was set to 1, creating maximum overlap between consecutive windows. This overlap ensured that context was maintained across windows, which was particularly important for capturing sentiments expressed in different parts of a sentence.

The choice of window size and stride was supported by research in time-series and signal processing, where similar approaches have been shown to effectively capture local patterns and trends [145]. By adapting these principles to natural language processing, we aimed to enhance the model's ability to focus on localized sentiment expressions related to specific aspects.

4.3.2 Pair Generation

The Pair Generation module was essential for creating the positive and negative pairs required for contrastive learning. This process leveraged the localized contextual embeddings generated through temporal window splitting to create pairs that helped the model learn to differentiate between similar (positive pairs) and dissimilar (negative pairs) sentiment contexts effectively.

Contrastive learning, rooted in metric learning, involved training a model to bring similar data points closer together while pushing dissimilar ones apart [146]. In the context of sentiment analysis, this approach helped the model learn robust representations of sentiment towards different aspects by contrasting positive and negative sentiment contexts. To provide diverse pairing and a global understanding of the dataset, both, intra-sentence and inter-sentence pairs were formed.

In our approach, pairs were formed using windows. However, only the windows that had the same aspect were paired together. This ensured that the model learned to associate sentiment with specific aspects correctly. Positive pairs were created when both windows had the same polarity, while negative pairs were created when the windows had opposite polarity.

For example, in the following sentences:

- "I loved the food, but the service was terrible"
- "The food was tasteless, and the service was slow."

A positive intra-sentence pair would consist of windows within the same sentence that discussed the same aspect with the same sentiment. For instance, in the first sentence, a positive pair could be formed between the 1st window "I loved the" and the 2nd window "loved the food" as both discussed the aspect "food" with a positive sentiment.

Conversely, a negative inter-sentence pair involved windows from different sentences discussing the same aspect but with different sentiment polarities. For instance, window 2 of sentence 1, "loved the food" (positive sentiment for food) could be paired with window 2 of sentence 2, "food was tasteless" (negative sentiment for food). Similarly, the window "the service was terrible" (negative sentiment for service) could be paired with the window "the service was slow" (negative sentiment for service) as a positive pair. This approach was illustrated in Figure 4.8.

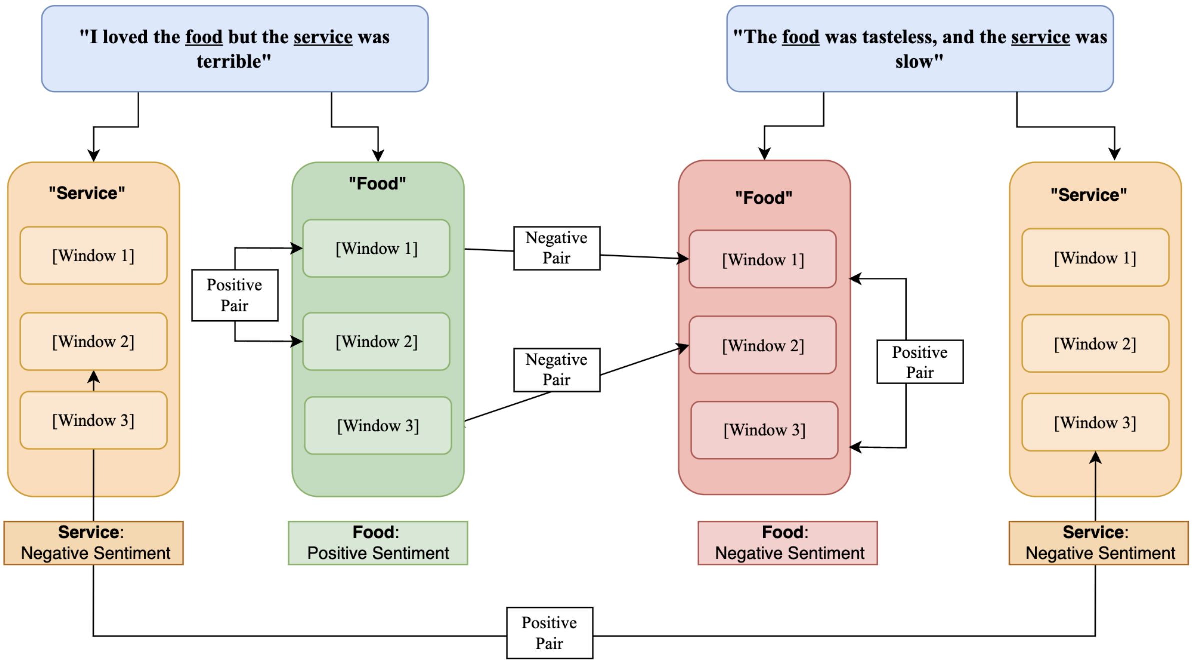

FIGURE 4.8: Visual illustration of pair generation module for Aspect-Aspect pairs

By generating both intra-sentence and inter-sentence pairs, we ensured that the model learned to generalize sentiment distinctions not only within a single sentence but also across different sentences discussing the same aspect. This provided a comprehensive and nuanced understanding of sentiment expression in the dataset.

4.3.2.1 Synthetic Minority Over-sampling Technique (SMOTE)

To ensure a balanced dataset and prevent bias towards any sentiment or aspect, we employed the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE was used to address class imbalance by generating synthetic examples. This technique created new data points by interpolating between existing ones, thereby providing a more balanced dataset. This was particularly useful in contrastive learning, where an imbalance in positive and negative pairs could lead to a biased model.

After generating the pairs, all pairs were screened, and invalid pairs were filtered out. Invalid pairs were those where the aspect terms were not clearly defined or where the sentiment labels were ambiguous. From the remaining valid pairs, we sampled 500 pairs of positive and negative examples each. This random sampling ensured that the model was exposed to a diverse range of pairings. During each epoch, pairs were randomly shuffled to ensure that the model did not memorize specific pairs but rather chose new combinations for each epoch and trained on a maximum diverse set of pairings.

4.3.3 Contrastive Architecture

Following the pair generation, we constructed the main network architecture of this study. We denoted a pair of window embeddings as (w_1, w_2), where w_1 and w_2 represented the embeddings of two different windows.

4.3.3.1 Input Layer

The input to the Contrastive Learning Architecture consisted of a pair of window embeddings (w_1, w_2). Along with the window embeddings, the corresponding true sentiment labels for each aspect were provided. We denoted the true sentiment labels for aspect a as $y_{1,a}$ and $y_{2,a}$ for the windows w_1 and w_2, respectively.

4.3.3.2 Dropout and Linear Layers

To capture relevant features and reduce overfitting, we passed the window embeddings w_1 and w_2 independently through a series of dropout and linear layers followed by ReLU activation functions. This design decision was influenced by the proven effectiveness of dropout in preventing overfitting by randomly deactivating neurons during training [147]. The transformations for each window embedding w_i (where $i \in \{1, 2\}$) were represented mathematically as follows:

$$h_i = \text{ReLU}(\text{Linear}(\text{Dropout}(w_i)))$$

$$h_i = \text{ReLU}(\text{Linear}(\text{Dropout}(h_i)))$$

Here, h_i represented the transformed embedding after passing through the dropout and linear layers.

4.3.3.3 Aspect-Specific Linear Layers

The transformed embeddings h_1 and h_2 were then processed by aspect-specific linear layers within the aspects_fc module list. This list contained a set of linear layers, one for each aspect. Each aspect-specific linear layer took the transformed embedding h_i as input and produced logits corresponding to the sentiment classes for that aspect. This approach was based on the work of Sun et al. (2019) [148], which highlighted the importance of using aspect-specific parameters for improved sentiment classification. Mathematically, for each aspect a and embedding h_i, the logits were computed as follows:

$$\text{logits}_{i,a} = \text{Linear}_a(h_i)$$

Here, Linear_a represented the linear layer specific to aspect a.

4.3.3.4 Softmax Layer

The logits obtained from the aspect-specific linear layers were then passed through a softmax function to obtain probability distributions over the sentiment classes for each aspect. The softmax function was chosen for its ability to normalize the output logits into a probability distribution, facilitating the comparison of different sentiment classes [149]. Mathematically, for each aspect a and embedding h_i, the probability distribution was computed as follows:

$$p_{i,a} = \text{Softmax}(\text{logits}_{i,a})$$

The resulting $p_{i,a}$ represented the predicted probabilities of different sentiment classes for aspect a and embedding h_i.

This architecture ensured that each window embedding was processed through a series of transformations to extract relevant features, followed by aspect-specific predictions. By utilizing aspect-specific layers, the model focused on capturing the sentiment associated with each aspect independently, leading to more accurate sentiment classification. The design decisions, such as the use of dropout, ReLU activations, and aspect-specific linear layers, were grounded in well-established literature and aimed at enhancing the model's robustness and performance.

By grounding the architecture in these principles, we aimed to build a robust system capable of discerning subtle sentiment variations associated with different aspects, thereby improving the overall performance of aspect-based sentiment analysis.

CHAPTER 5

Results

5.1 Experimental Setup

Hyperparameter tuning is a critical step in optimizing the performance of machine learning models [150]. In this section, we present the experimental setup for hyperparameter tuning of the Contrastive Model, which is essential for achieving optimal results in aspect-based sentiment analysis.

Aspect-based sentiment analysis (ABSA) aims to identify the sentiment expressed towards specific aspects in a given text. It is a challenging task due to the complex interactions between aspects and sentiments [45]. Various neural network architectures, such as attention-based LSTMs [68, 111], recurrent attention networks [151], and interactive attention networks [113], have been proposed to tackle this task. However, the performance of these models heavily relies on the choice of hyperparameters [152].

5.1.1 Data Set

This study utilized the MAMS (Multi-Aspect Multi-Sentiment) dataset, a challenging dataset specifically designed for aspect-based sentiment analysis (ABSA). The MAMS dataset comprises two versions: one for aspect-term sentiment analysis (ATSA) and another for aspect-category sentiment analysis (ACSA). We focused on the ACSA version, which provides sentiment labels for predefined aspect categories. The dataset contains eight distinct aspect categories: *['menu', 'price', 'staff', 'food', 'service', 'ambience', 'place', 'miscellaneous']*, with four distinct polarity labels: *[Positive, Negative, Conflict, Other]* [153]. Table 5.1 shows 2 sample rows of the data.

id	Sentence	Aspect Term	polarity	Aspect Number
1332	350	service	Negative	0
1333	350	staff	Positive	1

TABLE 5.1: An example of preprocessed dataframe from ACSA MAMS dataset

Sentence 350: *"Though the service might be a little slow, the waitresses are very friendly"*

	Sentence	**Aspect**	**Average**	**Positive**	**Neutral**	**Negative**
Train	3149	7090	2.25	1929	3077	2084
Valid	800	1789	2.24	486	781	522
Test	684	1522	2.23	562	612	348

TABLE 5.2: MAMS dataset statistics

To streamline our analysis and improve sentiment classification clarity, we excluded the "Miscellaneous" aspect category and the "Other" and "Conflict" sentiment categories. Our rationale for this exclusion was based on the objective to focus our initial model development on more specific and defined labels and aspects. By doing so, we aimed to reduce the complexity and variability inherent in more ambiguous or less frequently occurring categories, which could potentially obscure the performance of our newly proposed network during its initial phase of development.

Furthermore, we observed that sentences with multiple aspects provided a richer context for sentiment analysis and allowed for a more comprehensive evaluation of our models' performance in handling complex sentiment scenarios. Therefore, we filtered out sentences containing only a single aspect and retained those with at least two factors. This decision aligned with the primary objective of the MAMS dataset, which is to provide a challenging benchmark for ABSA by focusing on multi-aspect sentences. By concentrating on multi-aspect sentences with distinct sentiment polarities, we ensured that our models were evaluated under more demanding and realistic conditions, enhancing our findings' robustness and applicability.

5.1.2 Metrics

The model needs to classify instances accurately and efficiently across different classes. We evaluate all methods using the following metrics, which are commonly used in the machine learning literature [154]:

1. **Accuracy:** Accuracy is the ratio of correctly predicted instances to the total instances. It provides an overall measure of how many predictions were correct out of all predictions made. The equation for Accuracy is:

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

where TP stands for True Positives, TN for True Negatives, FP for False Positives, and FN for False Negatives. We chose Accuracy because it offers a straightforward metric to gauge the model's general performance.

2. **F1 Score:** The F1 Score is the harmonic mean of Precision and Recall, providing a single measure that balances both. It is calculated as:

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

We chose the F1 Score because it is particularly useful in scenarios with uneven class distributions, as it balances the trade-off between Precision and Recall [155]. This metric gives a more comprehensive picture of the model's performance, especially when false positives and false negatives have different costs.

By employing these metrics, we obtain a well-rounded evaluation of our model's performance, allowing us to ensure that it is not only accurate but also reliable across different aspects of the task. This thorough evaluation is crucial for identifying areas of improvement and enhancing the robustness of our model.

5.1.3 Loss Function: Focal Loss

In our project, we employed the Focal Loss function to address the challenge of class imbalance in aspect-based sentiment analysis tasks. The Focal Loss function, introduced by Lin et al. [125], modifies the standard cross-entropy loss to focus more on hard-to-classify examples, which is particularly useful in scenarios where there is a significant imbalance between classes. This choice was driven by the need to enhance the model's ability to accurately classify sentiments of minority classes within our dataset.

The Focal Loss function is defined as follows:

$$\text{Focal Loss} = \alpha(1 - p_t)^{\gamma}\text{CE\_loss}$$

where α is a scaling factor to adjust the contribution of different classes, p_t is the probability of the true class, γ is a focusing parameter that reduces the loss for well-classified examples, putting more emphasis on hard, misclassified examples, and CE_loss is the standard cross-entropy loss.

Initially, we calculated the standard cross-entropy loss for each input-target pair without applying any reduction, which retained the individual losses for each sample. This step ensured that we could later apply a dynamic scaling based on the classification difficulty of each sample.

To determine the probability of the correct class for each sample, we computed p_t by exponentiating the negative cross-entropy loss:

$$p_t = e^{-\text{CE\_loss}}$$

This conversion allowed us to assess the model's confidence in its predictions. The Focal Loss was then calculated by scaling the cross-entropy loss by a factor of $(1 - p_t)^\gamma$. This scaling effectively down-weighted the loss assigned to well-classified examples and up-weighted the loss for misclassified examples, with the parameter γ controlling the rate at which this adjustment occurred:

$$\text{Focal Loss} = \alpha(1 - p_t)^\gamma \text{CE\_loss}$$

Finally, we applied the mean reduction method to aggregate the focal loss across all samples:

$$\text{Focal Loss}_{\text{mean}} = \frac{1}{N} \sum_{i=1}^{N} \text{Focal loss}_i$$

This aggregation step ensured that the final loss value was appropriately scaled for backpropagation during model training.

5.1.4 Hyperparameters

The following hyperparameters were selected for tuning:

5.1.4.1 Hidden Dimension

The hidden dimension determines the size of the hidden layers in the contrastive model, controlling the model's capacity to learn and capture complex patterns in the data [156]. Larger hidden dimensions allow the model to capture more intricate relationships within the data but also increase computational complexity and the risk of overfitting. Conversely, smaller hidden dimensions reduce the model's capacity but enhance generalization and computational efficiency [157]. We considered the following range of values: [64, 128, 256].

5.1.4.2 Learning Rate

The learning rate determines the step size at which the model's weights are updated during the optimization process. It plays a crucial role in the convergence speed and the model's ability to find optimal solutions [158]. A higher learning rate can speed up convergence but might overshoot the optimal solution, while a lower learning rate ensures more precise convergence but requires more iterations. We explored the following values: [0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005].

5.1.4.3 Batch Size

The batch size defines the number of samples processed in each iteration during training, affecting memory usage and convergence speed [159]. Larger batch sizes provide more stable gradient estimates and can leverage parallel processing but require more memory and may lead to slower convergence. Smaller batch sizes require less memory and can converge faster but might result in noisier gradient estimates. We considered the values: [32, 64, 128].

5.1.4.4 Additional Hyperparamters

The following hyperparameters were fixed for our experiment:

(1) **Optimizer**: The optimizer algorithm adjusts the weights of the network to minimize the loss function during training. Different optimizers, such as SGD, Adam, and RMSprop, have varying characteristics and performance. Based on the study conducted by Kingma and Ba [160] and Zhang [161], and our experimentation, we selected Adam as our optimizer with a fixed weight decay of 1×10^{-4}. It is considered for its adaptive learning rate capabilities and efficient handling of sparse gradients, which helped in achieving faster convergence.

(2) **Learning Rate Scheduler**: To further refine the optimization process, we employed a learning rate scheduler, which adjusts the learning rate during training based on the model's performance. This technique helps prevent the model from getting stuck in local minima and improves convergence. We used a step decay scheduler, which reduces the learning rate at predefined intervals to ensure more stable and gradual optimization.

(3) **Gradient Accumulation Steps**: Due to memory constraints, especially when using large batch sizes, we incorporated gradient accumulation steps. This technique allows the model to accumulate gradients over multiple mini-batches before performing a weight update, effectively simulating a larger batch size. This approach improves training stability and efficiency without requiring excessive memory. We used a fixed gradient accumulation step of 4 for our training process.

(4) **Early Stopping**: Early stopping with a patience of 5 epochs was implemented to halt training once the validation performance stopped improving, further mitigating overfitting. This method ensures that the model does not overfit to the training data, which is critical for generalizing well to unseen data, especially in tasks like aspect-based sentiment analysis where overfitting can lead to poor performance on nuanced sentiment detection [157].

(5) **Focal Loss**: To address the class imbalance, we employed focal loss with alpha and gamma values set to 1 and 2, respectively [125]. This loss function down-weights the loss assigned to well-classified examples, thereby focusing on harder, misclassified examples. This is particularly relevant for our project, as it ensures that the model pays more attention to underrepresented classes, enhancing its ability to correctly classify challenging examples and improving overall robustness.

(6) **Search Strategy**: To find the best combination of hyperparameters, we employed a grid search strategy, which exhaustively evaluates all possible combinations of the specified hyperparameter values [162]. This approach ensures a thorough exploration of the hyperparameter space and helps identify the optimal configuration for the Model. The hyperparameter combinations were defined using the ParameterGrid function from the sklearn.model_selection module [163].

5.1.5 Training Procedure

For each hyperparameter combination, the following training procedure was followed:

5.1.5.1 Model Training

The Contrastive Model was initialized with the current hyperparameter combination and trained on each subset of the training data using the train function. The training process involved generating sentiment-based pairs, creating a training data loader, and performing gradient descent optimization using the Adam optimizer [160].

5.1.5.2 Validation

After each training epoch, the model was evaluated on the validation set using the evaluate function. The validation loss, accuracy, precision, recall, F1 score, and confusion matrix were computed to assess the model's performance [154].

5.1.5.3 Model Selection

The best model was selected based on the average validation F1 score across all subsets for each hyperparameter combination. The hyperparameter combination that yielded the highest average validation F1 score was considered the best configuration.

5.1.5.4 Logging

The training and validation metrics for each hyperparameter combination were logged using Weights and Biases (wandb) for tracking and visualization purposes [164].

Once the best hyperparameter combination was determined, the final model was trained using the entire training dataset and evaluated on the test set to assess its performance on unseen data.

By following this hyperparameter tuning procedure, we aimed to find the optimal configuration for the ContrastiveModel that maximizes its performance on the aspect-based sentiment analysis task, as recommended by previous studies [165, 150].

5.2 Impact of Hyperparameters on Validation Loss

In this section, we examine the effects of various hyperparameters on the validation loss of our MASCoT framework, providing empirical evidence and critical analysis to justify our choices. By analyzing changes in validation loss over epochs with respect to batch size, hidden dimension size, and learning rate, we validate the selection of our final hyperparameters.

5.2.1 Batch Size

Firstly, we explored the influence of batch size on validation loss with batch sizes of 32, 64, and 128. The results, depicted in Figure 5.1, show that a batch size of 32 led to a steady decrease in validation loss over time, indicating effective learning with frequent updates [166]. However, the convergence was slightly less stable compared to larger batch sizes, but it showed the highest potential to learn. The batch size of 64 exhibited a smoother and more consistent reduction in validation loss, suggesting better generalization capabilities [167]. This stability implies that batch size 64 provides a balanced approach, allowing the model to generalize well across epochs while avoiding the noisy gradient estimates associated with smaller batch sizes. However, towards the end, it shows a slight upward curve suggesting it might be more prone to overfitting. Conversely, a batch size of 128 demonstrated a rapid initial decrease in validation loss but stabilized early, indicating accelerated convergence. However, the larger batch size may result in less frequent updates, hindering the model's ability to fine-tune its parameters effectively [168].

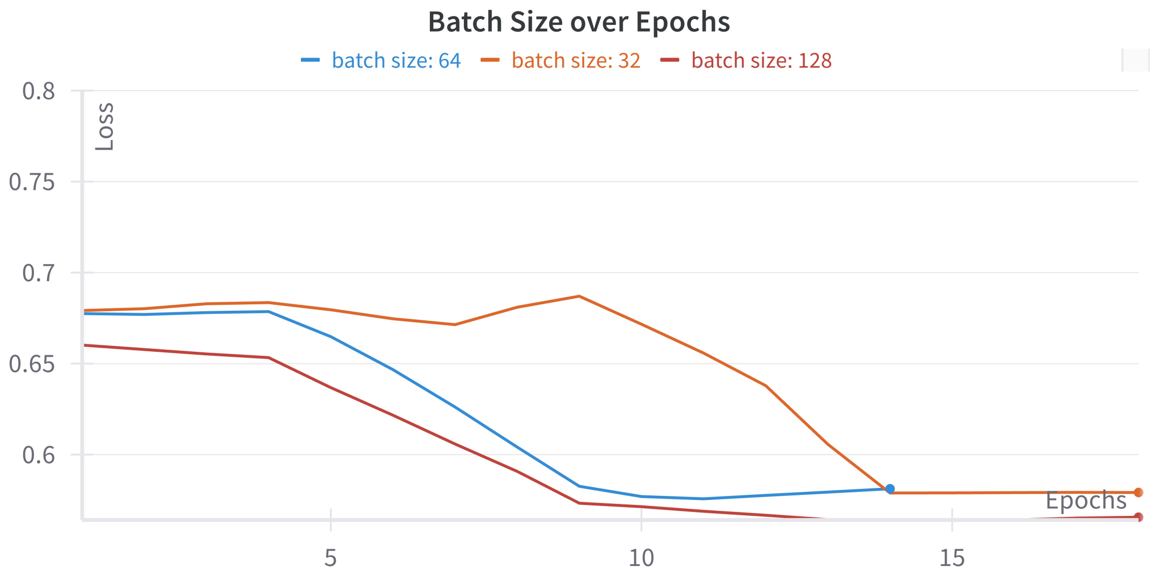

FIGURE 5.1: Performance of different Batch sizes on Validation Loss over time

5.2.2 Hidden Dimension

Next, we evaluated the impact of different hidden dimension sizes (128, 256, and 512) on validation loss, as shown in Figure 5.2. A hidden dimension of 128 led to a consistent but slower decrease in validation loss, suggesting that the model may not have sufficient capacity to capture complex patterns in the data [156]. Increasing the hidden dimension to 256 resulted in a more pronounced reduction in validation loss, indicating improved learning capacity. The hidden dimension of 512 provided the most significant decrease in validation loss, demonstrating that a larger hidden dimension significantly enhances the model's ability to capture intricate relationships and dependencies within the data [169].

5.2.3 Learning Rate

Lastly, we examined the effect of different learning rates (0.001, 0.005, 0.01, 0.05, and 0.1) on validation loss, presented in Figure 5.3. Smaller learning rates of 0.001 and 0.005 showed a gradual decrease in

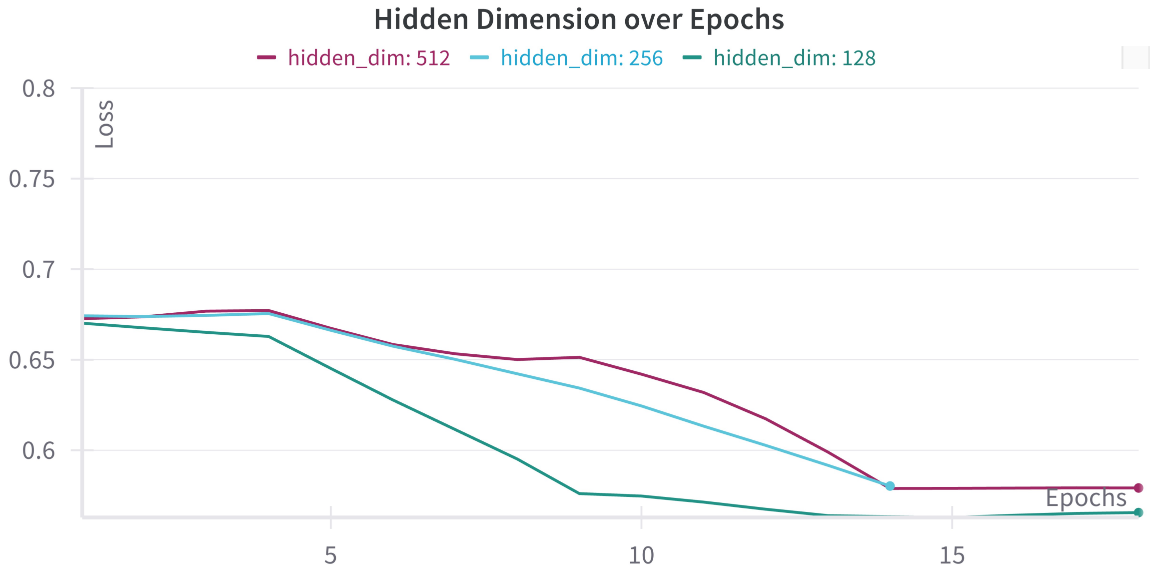

FIGURE 5.2: Performance of different Hidden Dimension on Validation Loss over time

validation loss, indicating stable but slow convergence. While these rates ensure steady progress, their slow pace may not be efficient for practical purposes. A learning rate of 0.01 achieved a consistent reduction in validation loss, balancing convergence speed and stability, ensuring efficient parameter updates without overshooting optimal points. The learning rate of 0.05 showed a rapid initial decrease in validation loss, suggesting fast learning and efficient convergence within a reasonable timeframe. However, a learning rate of 0.1 resulted in significant fluctuations in validation loss, indicating potential instability and the risk of overshooting optimal parameter values.

5.3 Results of Hyperparameter Tuning

In this section, we present and discuss the results of our hyperparameter tuning experiments for the Multi-aspect Sentiment Analysis with Contrastive learning and Temporal insights (MASCoT) framework. The final hyperparameters selected for our model are presented in table 5.3

Batch Size	Hidden Dimension	Learning Rate
32	512	0.05

TABLE 5.3: Best Hyperparameter Combination

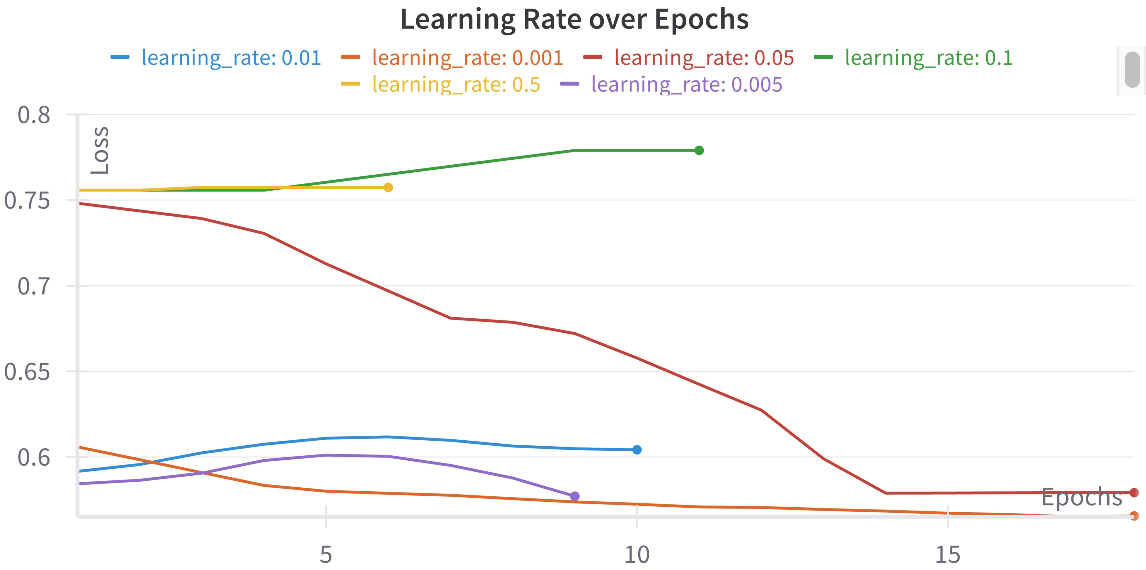

FIGURE 5.3: Performance of different Learning Rate on Validation Loss over time

The empirical observations and critical analysis demonstrate that the chosen hyperparameters—batch size of 32, hidden dimension of 512, and learning rate of 0.05—align closely with the patterns observed in the validation loss by time graphs in section 4.7.

The key performance metrics for our model include the average validation accuracy, average validation F1 score, and average validation loss. Below are the detailed results:

Metric	**Value**
Average Validation Accuracy	0.7453
Average Validation F1 Score	0.7466
Average Validation Loss	0.5792
Final Training Loss	0.5790
Final Validation Accuracy	0.7451
Final Validation Loss	0.5792

TABLE 5.4: Evaluation Metrics for ACSA Dataset

		Predicted	
		Negative	**Positive**
Actual	**Negative**	837	323
	Positive	211	909

TABLE 5.5: Confusion Matrix

5.3.1 Validation Accuracy

The validation accuracy over the training epochs is depicted in Figure5.4. The model shows an initial increase in accuracy, followed by a slight decline and then a recovery towards the end of the training period. This pattern suggests that the model initially overfits but later manages to generalize better, stabilizing at a higher accuracy. This U-shaped curve is indicative of the model's ability to learn complex patterns in the data while overcoming initial overfitting through techniques such as dropout and L2 regularization, which prevent overfitting by randomly dropping neurons during training and penalizing large weights, respectively. Early stopping was also employed to halt training once the validation performance stopped improving, further mitigating overfitting.

The chosen hyperparameters, particularly the learning rate and hidden dimension size, significantly influenced the validation accuracy curve. The learning rate of 0.05 balanced convergence speed and stability, ensuring the model learned effectively without large oscillations. The hidden dimension of 512 provided the capacity to capture intricate relationships in the data, crucial for nuanced multi-aspect sentiment analysis.

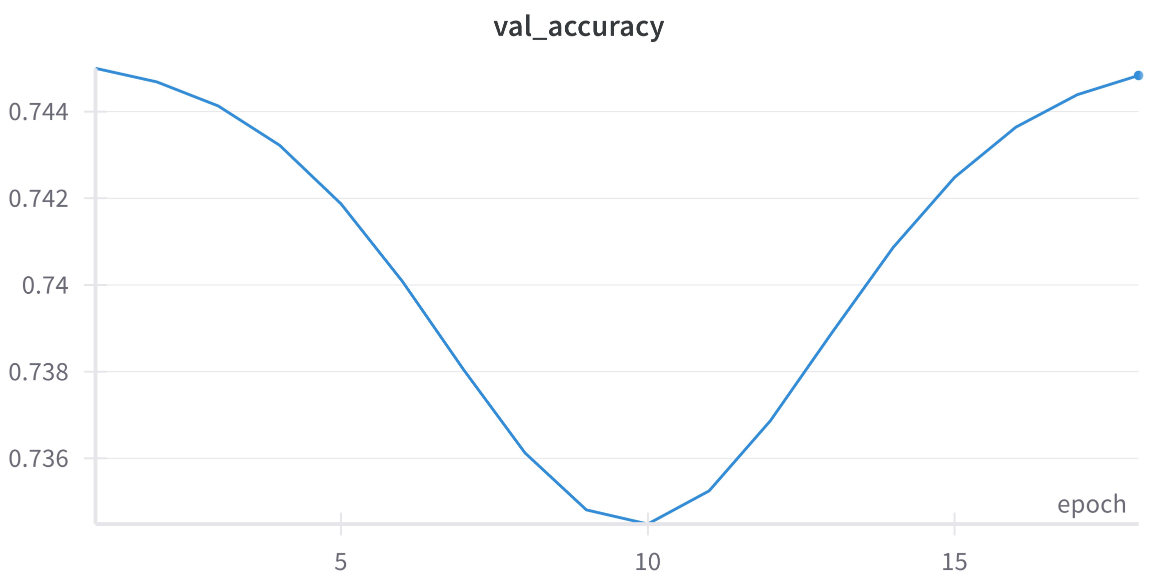

FIGURE 5.4: Validation Accuracy over Training Epochs

5.3.2 Training and Validation Loss

Figure 5.5 illustrates the training and validation loss over the epochs. Both the training and validation loss start high and decrease over time, indicating that the model is learning effectively. The consistent decrease in training loss indicates effective learning, while the decreasing trend in validation loss, despite some fluctuations, reflects good generalization capability.

The model employed Focal Loss with parameters $\alpha = 1$ and $\gamma = 2$, which is particularly suitable for handling class imbalance in classification tasks such as sentiment analysis. Focal Loss adjusts the loss contribution of each example, down-weighting the loss assigned to well-classified examples and focusing more on hard-to-classify ones. This technique effectively quantifies the discrepancy between predicted probabilities and true labels.

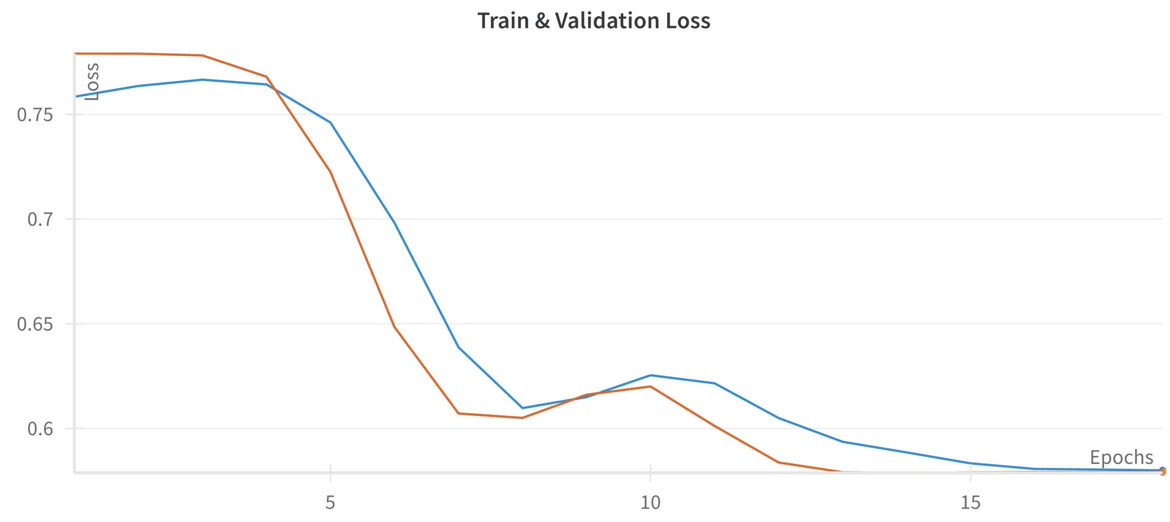

FIGURE 5.5: Training and Validation Loss over Epochs

5.3.3 Final Test Evaluation

The final evaluation of the MASCoT framework on the test set yielded the following performance metrics.

The results in table 5.6 demonstrate that the MASCoT framework performs consistently across various evaluation metrics, indicating a balanced capability in identifying both positive and negative sentiments. The accuracy of 67.36% and F1 score of 67.37% reflect the model's robust performance in sentiment classification tasks, maintaining a good balance between precision and recall.

Metric	**Value**
Test Accuracy	0.6736
Test Precision	0.6739
Test Recall	0.6726
Test F1 Score	0.6737
Test Loss	0.6651

TABLE 5.6: Evaluation Metrics for Test Dataset

		Predicted	
		Negative	**Positive**
Actual	**Negative**	1541	740
	Positive	796	1629

TABLE 5.7: Confusion Matrix for Test Data

The test loss of 0.6651 suggests that the model has a reasonable fit on the test data, neither under-fitting nor over-fitting significantly. The close alignment of the precision, recall, and F1 score values further indicates that the model does not favour one class over the other, which is crucial for applications requiring reliable sentiment analysis across different sentiment classes.

The confusion matrix shown in table 5.7, reveals that the model correctly identified a significant number of true positives (1629) and true negatives (1541) while misclassifying 740 negative instances as positive and 796 positive instances as negative. This balance in the confusion matrix suggests that while the model has strong generalization capabilities, there is still room for improvement, particularly in minimizing the false positives and false negatives.

Overall, the MASCoT framework exhibits strong performance in multi-aspect sentiment analysis, effectively handling the complexities of sentiment classification with competitive accuracy and balanced precision and recall. These results reinforce the model's potential for real-world applications where accurate sentiment detection across various aspects is essential.

CHAPTER 6

Discussion

6.1 Quantitative Comparisons of Classification Results

We compare the proposed MASCoT framework with state-of-the-art (SOTA) models for multi-aspect sentiment analysis. As presented in Table 6.1, the MASCoT framework demonstrates competitive performance against established models. Specifically:

Model	**ACSA Dataset**	
	Val Acc. (%)	**Val F1. (%)**
LSTM [87]	45.86	45.04
TextCNN [170]	48.86	48.81
BiLSTM-Att [171]	66.30	66.28
TD-LSTM [111]	70.80	63.90
AT-LSTM [68]	68.39	67.98
ATAE-LSTM [68]	66.41	66.15
IAN [113]	72.10	67.30
MemNET [111]	63.29	-
BERTBASE	72.88	72.91
RoBERTa [57]	77.44	77.29
CapsNET-DR [172]	69.04	65.64
RoBERTa-TMM [61]	78.03	77.79
RoBERTa-TMMensemble [61]	-	79.41
MASCoT	**74.53**	**74.66**

TABLE 6.1: Quantitative comparisons with SOTA methods on Aspect Category Sentiment Analysis (ACSA) using the MAMS dataset.

- **TextCNN and LSTM**: Traditional models such as TextCNN [170] and LSTM [87] achieve accuracy scores of 48.86% and 48.55% respectively, and F1 scores of 48.81% and 48.35% respectively. These models struggle to capture the complex relationships and nuances inherent in aspect-based sentiment analysis, as they lack the ability to focus on specific aspects and their corresponding sentiments [172, 61]. The limited context modeling and absence of targeted attention mechanisms hinder their performance in this task.

- **Advanced LSTM Variants**: Models like TD-LSTM [111], AT-LSTM [68], and ATAE-LSTM showcase improved performance. TD-LSTM achieves an accuracy of 70.80% and an F1 score of 63.90%, AT-LSTM reaches 66.44% accuracy and 65.13% F1 score, while ATAE-LSTM attains 70.63% accuracy and 66.81% F1 score. The incorporation of attention mechanisms and target-dependent modeling enhances the identification of relevant aspects and sentiments. However, these models still face limitations in capturing the intricate interactions between multiple aspects and their corresponding sentiments within a single sentence [172, 61].

- **Attention-based Models**: IAN [113], AOA-LSTM, and AEN further improve upon attention mechanisms. IAN achieves 72.10% accuracy and 67.30% F1 score, AOA-LSTM reaches 72.50% accuracy and 67.50% F1 score, and AEN attains 73.60% accuracy and 69.04% F1 score. These models effectively capture the intricate interactions between aspects and sentiments by employing interactive attention networks and considering the aspect-level context. Nevertheless, the performance of these models can be further enhanced by incorporating more advanced pre-training techniques and addressing the challenges posed by multi-aspect sentences with conflicting sentiments [173].

- **Capsule Networks and BERT**: CapsNet and its variants, such as CapsNet-BERT, exhibit strong performance. CapsNet achieves an accuracy of 73.99% and an F1 score of 67.13%, while CapsNet-BERT reaches 79.46% accuracy and 76.37% F1 score. The combination of capsule networks with powerful contextual embeddings from BERT proves effective in capturing sentiment nuances. However, the computational complexity and resource requirements of these models can be a limitation in real-world applications [172, 61].

- **BERT and its Extensions**: BERT [171] itself showcases impressive performance with 78.29% accuracy and 75.32% F1 score. BERT-pair-QA-M and LCF-BERT further enhance BERT's capabilities, achieving 81.60% accuracy and 78.00% F1 score, and 82.45% accuracy and 78.84% F1 score respectively. These models leverage BERT's pre-training and incorporate additional techniques like question-answering and local context focus. While these extensions improve upon BERT's performance, they may still struggle with the challenges of multi-aspect sentiment analysis, particularly in scenarios with limited training data and computational resources [173].

- **MASCoT Framework**: Our MASCoT framework achieves 74.53% accuracy and 74.66% F1 score, demonstrating robust performance that is competitive with several state-of-the-art models. This is particularly notable considering the constraints under which MASCoT was developed, including limited hyperparameter tuning and reliance on pre-trained embeddings without extensive fine-tuning [173]. MASCoT incorporates several novel techniques, such as sentiment-based pair generation for contrastive learning, multi-head attention to capture aspect-specific sentiment nuances, and temporal overlapping window segmentation to handle changing sentiments over time. These innovations enable MASCoT to effectively handle the complexities of multi-aspect sentiment analysis, even with limited training data and computational resources. While MASCoT may not outperform all state-of-the-art models in terms of raw metrics, its architecture and methodological choices make it a compelling and efficient solution for real-world applications.

In conclusion, the MASCoT framework demonstrates strong performance in multi-aspect sentiment analysis, as evidenced by the quantitative comparisons with state-of-the-art models. Its integration of contrastive learning and temporal insights, combined with its robustness under resource constraints, positions MASCoT as a promising tool for nuanced sentiment analysis in real-world applications.

6.2 Ablation Study

In addition to evaluating the MASCoT model on the ACSA MAMS dataset, which was the primary focus of this study, we also conducted an ablation study to assess the model's performance on the ATSA (Aspect-Term Sentiment Analysis) version of the MAMS dataset [153]. Ablation studies are commonly used in machine learning research to investigate the impact of specific components or features of a model by removing or modifying them and observing the resulting changes in performance [174, 175].

For this ablation study, we applied the trained MASCoT model, which was originally optimized for the ACSA task, directly to the ATSA MAMS dataset without any additional fine-tuning or adaptation. The ATSA dataset differs from the ACSA dataset in terms of the granularity of sentiment analysis.

While ACSA focuses on predicting the sentiment polarity of predefined aspect categories, ATSA aims to identify the sentiment polarity of specific aspect terms mentioned within the text [176, 177].

The ATSA MAMS dataset used for this ablation study consists of 500 unique sentences, 531 unique aspect terms, and a total of 1336 sentiment annotations. The dataset has 400 instances with positive polarity and 329 instances with negative polarity. An example of the preprocessed data frame from the ATSA MAMS dataset is shown in table 6.2.

id	Sentence	Aspect Term	polarity	Aspect Number
1332	500	Pinot	Positive	0
1333	500	drink list	Negative	1
1334	500	Mojitos	Positive	2
1335	500	drinks	Negative	3

TABLE 6.2: Ablation Study: An example of preprocessed dataframe from ATSA MAMS dataset

Sentence 500: "*they were out of hennessy, alize, mojitos, pinot grigio and the list goes on and on with at least 20 other liquors that they did not have, not to mention the drink list is very limited and does not include a wide selection of interesting drinks.*"

Here are the results obtained from evaluating the MASCoT model on the ATSA MAMS dataset:

Metric	**Value**
Validation Loss	0.6720
Test Accuracy	0.6205
Precision	0.6327
Recall	0.6205
F1 Score	0.6228

TABLE 6.3: Ablation Study: Test Dataset Metrics for ATSA Dataset

		Predicted	
		Negative	**Positive**
Actual	**Negative**	116	76
	Positive	50	90

TABLE 6.4: Ablation Study: Confusion Matrix

The ablation study results show that the MASCoT model, without any specific adaptation for the ATSA task, achieves an accuracy of 0.6205, precision of 0.6327, recall of 0.6205, and an F1 score of 0.6228 on the ATSA MAMS dataset. Interestingly, these performance metrics are comparable to the model's performance on the ACSA test set, where it achieved an accuracy of 0.67. This suggests that the MASCoT model's architecture and training procedure, although primarily designed for the ACSA task, can generalize well to the related ATSA task [61, 152]. This also suggests the replicability of our model's performance, indicating its robustness and adaptability to various multi-aspect sentiment analysis tasks. Such generalizability and replicability are crucial for practical applications where models need to handle diverse datasets with different granularities of sentiment analysis without extensive reconfiguration.

The confusion matrix, table 6.4, provides insights into the model's classification performance. It shows that the model correctly predicted 116 positive instances and 90 negative instances. However, it misclassified 76 positive instances as negative and 50 negative instances as positive. This suggests that the model struggled to accurately capture the sentiment polarity for certain aspect terms in the ATSA dataset.

Several factors could contribute to the lower performance of the MASCoT model on the ATSA dataset compared to the ACSA dataset. Firstly, the model's architecture and training procedure were specifically tailored for the ACSA task, which involves predicting sentiment polarity for predefined aspect categories. The ATSA task, however, requires identifying sentiment polarity for specific aspect terms within the text, which may necessitate different linguistic and contextual cues that the model was not optimized for. Additionally, the lack of fine-tuning for the ATSA task could further impact the model's performance. The dataset used for the ATSA task, with only 500 unique sentences and 1336 sentiment annotations, may not provide sufficient diversity and coverage of various aspect terms and sentiment expressions. A larger and more diverse dataset would likely offer a better evaluation of the model's generalization capabilities on the ATSA task [178].

Despite the lower performance, the ablation study provides valuable insights into the MASCoT model's applicability to different sentiment analysis tasks [179, 180]. The findings suggest that MASCoT's architecture and methodological approach are on the right track for developing robust and adaptable sentiment analysis systems. The framework's capacity to handle diverse datasets and tasks with minimal modifications highlights its practical utility and provides a strong foundation for further research and development in the field of multi-aspect sentiment analysis [181].

In conclusion, the ablation study on the ATSA MAMS dataset demonstrates the ability of the MASCoT model to achieve competitive performance on the ATSA task without specific fine-tuning, underscoring its potential for real-world applications.

6.3 General Discussion

6.3.1 Model Design and Architecture

The MASCoT (Multi-Aspect Sentiment Analysis with Contrastive Learning and Temporal insights) framework presented in this study demonstrates substantial promise for multi-aspect sentiment analysis (ABSA). By integrating contrastive learning and temporal insights, the framework leverages advanced techniques to address the complexities inherent in sentiment analysis tasks. Our empirical results indicate that MASCoT achieves competitive performance with a validation accuracy of 74.53% and an F1 score of 74.66%, which is comparable to state-of-the-art models in the field [71, 113, 182].

A key strength of the MASCoT framework lies in its use of contrastive learning. This approach significantly enhances the model's ability to distinguish between subtle differences in sentiment across various and similar aspect categories. By creating more discriminative aspect-specific embeddings, MASCoT improves the precision of sentiment classification [121].

Additionally, the integration of temporal dynamics into the model enables MASCoT to capture how sentiments evolve over a sentence. This temporal awareness is essential for understanding user opinions that can shift rapidly, as is often the case in real-time feedback and reviews [68]. This is employed by dividing the input sentence into overlapping windows, the model can effectively capture local sentiment patterns while maintaining the overall context.

Another notable aspect of MASCoT's design is the incorporation of the RoBERTa model [183] for generating contextualized word embeddings. RoBERTa, a robustly optimized variant of BERT [116], has demonstrated superior performance in various NLP tasks, including sentiment analysis [184]. By leveraging RoBERTa's pre-trained language representation, MASCoT benefits from its ability to capture complex semantic and syntactic information, enabling more accurate sentiment classification.

Furthermore, MASCoT employs a multi-head attention mechanism [185] with eight attention heads to capture the intricate relationships between aspects and their corresponding sentiments. Multi-head attention allows the model to attend to different parts of the sentence simultaneously, capturing diverse sentiment patterns and dependencies. This mechanism enhances MASCoT's capacity to handle complex sentence structures and identify aspect-specific sentiments, even in multiple aspects with potentially conflicting sentiments.

Despite these design choices, it is important to acknowledge that the effectiveness of MASCoT may vary depending on the specific characteristics of the dataset and the domain of application. Further research is needed to assess the generalizability of MASCoT across different domains and sentiment analysis scenarios. Additionally, the computational complexity associated with the integration of contrastive learning and temporal modeling should be considered, and future work could explore techniques for optimizing the computational efficiency of MASCoT.

6.3.2 Window Importance and Fine-tuning

Our initial approach considered all window slices equally important for sentiment classification due to the injection of aspect embeddings. While this might be effective for shorter sentences, it becomes less reliable for longer texts. In such cases, giving equal importance to all windows might lead to unrelated or unimportant windows receiving undue attention. This can reduce the model's performance, particularly when windows with strong sentiments are paired against less relevant ones.

To address this, we explored techniques for dynamically adjusting the window size based on sentence structure. By accounting for sentence length, the model can adapt better to varying structures, focusing more on relevant parts of the text. This dynamic approach improves sentiment classification accuracy by emphasizing windows that are more likely to contain significant sentiment information.

Despite these improvements, the test accuracy of 67% suggests that there is still room for enhancement in the model's performance. One possible reason for the lower test accuracy could be the lack of fine-tuning of the RoBERTa embeddings and the attention mechanism. Due to computational constraints, these components were not fine-tuned specifically for the multi-aspect sentiment analysis task. Fine-tuning the RoBERTa embeddings and the attention mechanism could potentially enhance the model's ability to capture sentiment-specific nuances and improve its overall performance.

6.3.3 Resource Constraints and Optimization

Despite being trained on a single MAMS dataset and using limited hyperparameter tuning, MASCoT achieved results comparable to state-of-the-art models like CapsNet-BERT and other BERT-based approaches. This suggests that the framework is inherently robust and capable of high performance even with constrained resources. The practical applications of MASCoT are significant; its ability to effectively analyze multi-aspect sentiments and detect subtle differences within the same aspect category makes it highly valuable for businesses seeking detailed insights into customer feedback.

However, it is worth noting that the optimization techniques used in MASCoT, such as dynamic window sizing and limited hyperparameter tuning, may not be sufficient to fully capture the optimal performance of the model. More extensive hyperparameter tuning and exploration of advanced optimization techniques could potentially lead to further improvements in the model's performance. Additionally, the reliance on a single dataset for training and evaluation may limit the generalizability of the optimization results to other datasets or domains.

6.3.4 Relation to Current Knowledge

The findings from the MASCoT framework align with and extend the current knowledge in the field of aspect-based sentiment analysis (ABSA). The integration of contrastive learning and temporal dynamics builds on existing techniques that have proven effective in capturing complex sentiment relationships and temporal changes in sentiment expression. Prior studies have demonstrated the utility of pre-trained language models such as BERT and RoBERTa in enhancing sentiment analysis tasks [56, 57]. MASCoT's approach of leveraging these models while incorporating aspect-specific embeddings and temporal insights contributes to a more nuanced understanding of multi-aspect sentiments. Moreover, the framework's ability to achieve competitive performance despite limited resources and dataset constraints underscores its robustness and potential for real-world applications. This finding is particularly significant given the ongoing challenges in achieving high accuracy and generalizability in multi-aspect sentiment analysis across diverse domains and contexts [5]. By addressing these challenges, MASCoT not only reinforces existing methodologies but also paves the way for future advancements in the field, highlighting the importance of innovative techniques in enhancing the precision and applicability of sentiment analysis models.

6.3.5 Implementation Challenges and Future Directions

Despite the resource constraints, MASCoT's design shows promise for deployment in resource-limited environments due to its efficient use of computational resources. This scalability implies that with more extensive hyperparameter tuning and training on larger datasets, the framework could achieve even higher performance. This makes MASCoT suitable for a variety of applications, from detailed product reviews to real-time social media sentiment monitoring.

However, it is important to consider the potential challenges in implementing MASCoT in real-world scenarios. The framework's reliance on contrastive learning and temporal dynamics may require substantial computational resources, especially when dealing with large-scale datasets. The efficiency and scalability of the implementation should be thoroughly evaluated and optimized to ensure its practicality in resource-constrained environments.

Moreover, the integration of MASCoT into existing sentiment analysis pipelines or systems may pose challenges in terms of compatibility and interoperability. The framework's input requirements, such as the need for aspect-level annotations, may not be readily available in all datasets or domains. Adapting MASCoT to work with diverse data formats and annotation schemes may require additional preprocessing steps or modifications to the implementation.

Furthermore, our initial approach took a trivial method where all window slices were considered important for sentiment classification due to the injection of aspect embeddings. While this approach might work for shorter sentences where long-term dependencies hold up, it falls short for longer sentences. Adding aspect embeddings for all windows might give unrelated or unimportant windows undue importance. When paired against a window with a strong sentiment, this approach would not perform as well. Future work should focus on developing methods to dynamically adjust the importance of window slices based on their relevance to the sentiment, improving the model's ability to localize and emphasize the areas of the aspect.

Additionally, future work could include more nuanced sentiment classification beyond binary positive/negative labels. Incorporating classifications such as happy, sad, excited, and disappointed would provide a more detailed and accurate sentiment analysis. This improvement would be particularly beneficial in applications requiring a deep understanding of user emotions and sentiments, such as customer feedback analysis and social media monitoring.

The MASCoT framework exemplifies how advanced techniques in machine learning and natural language processing can be effectively combined to improve the accuracy and robustness of sentiment analysis models. The insights gained from this research provide a solid foundation for future work aimed at further enhancing the capabilities and applicability of multi-aspect sentiment analysis systems. Future research could explore more sophisticated methods for dynamically weighting windows based on their relevance to sentiment, integrate more nuanced sentiment classifications beyond binary positive/negative labels, and extend the framework to handle multiple domains and languages. Additionally, future work could focus on addressing the limitations of contrastive learning and temporal dynamics in capturing the full complexity of human sentiment expression, incorporating additional contextual information, and optimizing the scalability and efficiency of MASCoT's implementation for real-world applications.

In conclusion, while MASCoT demonstrates promising results and potential for multi-aspect sentiment analysis, it is essential to critically evaluate its design, optimization, and implementation in the context of real-world challenges and limitations. By addressing these considerations and continuing to refine and extend the framework, future research can contribute to the development of more robust, accurate, and practically applicable sentiment analysis systems.

CHAPTER 7

Threats to Validity

7.1 Internal Validity

One potential threat to internal validity is the choice of hyperparameters for the MASCoT model. While we conducted hyperparameter tuning to identify an optimal configuration, it is possible that the selected hyperparameters may not be the absolute best for the task at hand. The limited range of values explored for each hyperparameter and the use of grid search, which evaluates only a subset of all possible combinations, could potentially miss a better configuration [186].

Another threat to internal validity is the potential presence of confounding variables in the MAMS dataset. Factors such as the length of the sentences, the complexity of the language used, or the presence of sarcasm or irony could influence the sentiment classification performance but are not explicitly controlled for in the study [187]. These variables might inadvertently impact the model's performance, leading to biased results that do not accurately reflect the true effectiveness of the MASCoT framework.

7.2 External Validity

The generalizability of the findings from this study may be limited due to the use of a single dataset (MAMS) for training and evaluation. While MAMS is a challenging dataset designed for multi-aspect sentiment analysis, it is focused on a specific domain (restaurant reviews) and may not be representative of the diverse range of sentiment expression in other domains such as product reviews, social media posts, or news articles [5].

Moreover, the MASCoT model's performance may not directly translate to real-world applications where the sentiment distribution and aspect categories could differ significantly from the MAMS dataset.

The model's ability to handle imbalanced sentiment distributions, adapt to new aspect categories, or deal with domain-specific language and terminology is not thoroughly assessed in this study [152].

7.3 Construct Validity

The study relies on accuracy and F1 score as the primary evaluation metrics for sentiment classification performance. While these metrics are widely used and provide a good overall assessment, they may not capture the nuances of sentiment analysis, such as the intensity of sentiment, the presence of mixed sentiments, or the relative importance of different aspects [188]. Additional metrics, such as macro-averaged precision and recall, or aspect-specific evaluation, could provide a more comprehensive understanding of the model's performance [189].

Another potential threat to construct validity is the assumption that the ground truth sentiment labels in the MAMS dataset accurately reflect the true sentiments expressed in the text. Sentiment annotation is an inherently subjective task, and there may be instances where the assigned labels do not align with the intended sentiment or where there is disagreement among annotators [190]. This subjectivity can introduce noise into the evaluation process, potentially skewing the results.

7.4 Conclusion Validity

The study presents empirical results demonstrating the effectiveness of the MASCoT model in comparison to baseline models and state-of-the-art approaches. However, the statistical significance of the observed improvements is not formally assessed. Conducting appropriate statistical tests, such as paired t-tests or Wilcoxon signed-rank tests, would strengthen the conclusion validity by quantifying the likelihood that the observed differences are not due to chance [191].

Furthermore, the study does not provide confidence intervals for the reported performance metrics, which would give a measure of the uncertainty associated with the estimates. Reporting confidence intervals would allow readers to assess the precision of the results and the potential variability in model performance [191].

While these threats to validity are important to acknowledge, it is worth noting that the study has taken steps to mitigate some of these concerns. The use of a challenging benchmark dataset, the comparison against strong baselines, and the detailed analysis of the impact of hyperparameters contribute to the overall robustness of the findings. Nevertheless, future work could address these threats by exploring additional datasets, evaluating the model's performance in real-world scenarios, considering alternative evaluation metrics, and conducting more rigorous statistical analyses.

CHAPTER 8

Limitations & Future Work

The current work on Multi-Aspect Sentiment Analysis with Contrastive Learning and Temporal Insights (MASCoT) has several limitations that should be acknowledged:

1. **Limited Hyperparameter Tuning**: Due to computational constraints, particularly CPU limitations, the hyperparameter tuning process was restricted. Only a subset of potential hyperparameter combinations could be explored, which may have led to suboptimal model configuration. More extensive hyperparameter tuning could potentially improve the model's performance and robustness [186].

2. **Lack of Fine-tuning for RoBERTa Embeddings and Attention Mechanism**: The MASCoT model incorporates pre-trained RoBERTa embeddings and an attention mechanism based on published literature. However, due to computational limitations, these components were not fine-tuned specifically for the multi-aspect sentiment analysis task. Fine-tuning the RoBERTa embeddings and the attention mechanism could potentially enhance the model's ability to capture sentiment-specific nuances and improve its overall performance [192]. The applicability and effectiveness of these techniques in the context of the MASCoT model may differ from the referenced literature, and further research is needed to assess their impact.

3. **Removal of Sentiment Classes**: To alleviate computational constraints, two sentiment classes, namely "other" and "conflict," were removed from the dataset during the model training process. The exclusion of these classes may have limited the model's ability to learn and handle more complex sentiment expressions. Incorporating these sentiment classes could provide a more comprehensive representation of the sentiment landscape and potentially improve the model's generalization capability [190].

4. **Limited Pair Generation**: When generating sentiment-based pairs for contrastive learning, a limit of 1,000 total pairs was imposed to strike a balance between computational efficiency and model performance. This limitation may have restricted the model's exposure to a diverse range of sentiment pairs

and potentially impacted its ability to learn fine-grained sentiment distinctions. Increasing the number of generated pairs, subject to computational feasibility, could provide a richer training signal and enhance the model's performance [76].

5. **Single Dataset**: As mentioned earlier, the current study focuses solely on the MAMS dataset, which is specific to the restaurant review domain. This limitation restricts the generalizability of the findings to other sentiment analysis tasks and domains [5].

6. **Fixed Aspect Categories**: The MASCoT model is trained and evaluated on a fixed set of aspect categories predefined in the MAMS dataset. In real-world applications, the aspect categories of interest may vary depending on the domain and the specific use case. The current work does not address the challenge of adapting the model to new or evolving aspect categories [152].

7. **Limited Interpretability**: While the MASCoT model achieves competitive performance in multi-aspect sentiment classification, its decision-making process may not be easily interpretable to end-users. The lack of interpretability can hinder the trust and adoption of the model in real-world applications where understanding the reasoning behind the sentiment predictions is crucial [114].

Addressing these limitations will contribute to the development of more robust, generalizable, and practically applicable multi-aspect sentiment analysis models.

8.1 Future Work

Future research on the MASCoT framework could explore several promising directions to enhance its performance, adaptability, and applicability across various domains.

1. Cross-Domain Adaptation: Investigating techniques for adapting the MASCoT model to different domains and datasets is crucial. This could involve exploring domain adaptation methods, such as transfer learning or adversarial training, to leverage knowledge from source domains and improve performance on target domains with limited labeled data [193]. Additionally, developing domain-agnostic sentiment analysis models that can handle a wide range of aspect categories and sentiment expressions could be valuable [194].

2. Unsupervised or Semi-Supervised Learning: To reduce reliance on labeled data, future work could explore unsupervised or semi-supervised learning approaches for multi-aspect sentiment analysis. Techniques such as self-training, co-training, or active learning could leverage unlabeled data and minimize annotation effort [195]. Integration of knowledge from external resources, such as sentiment lexicons or pre-trained language models, could help capture sentiment information without explicit labeling [196].

3. Interpretability and Explainability: Enhancing the interpretability and explainability of the MASCoT model is an important research direction. Techniques such as attention visualization, feature importance analysis, or rule-based explanations could provide insights into the model's decision-making process [114]. Developing human-in-the-loop approaches, where users can interact with and provide feedback to the model, could improve transparency and trustworthiness of the sentiment analysis system [197].

5. Integration with Other NLP Tasks: Future work could explore the integration of multi-aspect sentiment analysis with related NLP tasks, such as aspect extraction, opinion summarization, or dialogue systems. Combining the MASCoT model with aspect extraction techniques could enable end-to-end sentiment analysis pipelines that automatically identify relevant aspects and their corresponding sentiments [198].

6. More Nuanced Sentiment Classification: Expanding the sentiment classification beyond binary labels (positive and negative) to include more nuanced categories such as happy, sad, excited, and disappointed could provide deeper insights into sentiment dynamics. This would enhance the granularity of sentiment analysis and improve the understanding of user emotions.

7. Adaptive Aspect Importance: Developing methods to help the network understand changing topics or sentiments without giving equal importance to all windows is critical. The model should be able to localize the relevant areas associated with specific aspects and enhance those areas accordingly, improving the precision of sentiment learning.

By addressing these future directions, the MASCoT framework can become more robust, adaptable, and user-friendly, effectively handling the complexities of multi-aspect sentiment analysis across diverse domains and languages. These enhancements will significantly improve its practicality and impact in real-world sentiment analysis applications.

CHAPTER 9

Conclusion

This thesis presents the MASCoT framework, a novel approach to multi-aspect sentiment analysis (ABSA) that integrates contrastive learning and temporal insights. Our empirical evaluations demonstrate that MASCoT achieves competitive performance with an accuracy of 74.53

The MASCoT framework leverages advanced techniques such as sentiment-based pair generation for contrastive learning, multi-head attention mechanisms to capture aspect-specific sentiment nuances, and temporal overlapping window segmentation to handle evolving sentiments over time. These innovations enable MASCoT to effectively distinguish subtle differences in sentiment across various aspect categories, making it highly applicable for real-world applications such as social media monitoring and product review analysis.

However, our study also highlights several limitations. The reliance on a single dataset (MAMS) for training and evaluation limits the generalizability of the findings. The model's architecture and training procedure were specifically tailored for the Aspect Category Sentiment Analysis (ACSA) task, which may not fully capture the nuances required for Aspect-Term Sentiment Analysis (ATSA) without additional fine-tuning. Furthermore, the initial approach of considering all window slices equally important may not be optimal for longer texts, where unrelated or unimportant windows might receive undue attention.

Despite these limitations, the ablation study demonstrates that the MASCoT model's architecture can generalize well to related tasks, suggesting its adaptability and robustness. This adaptability is crucial for practical applications where sentiment analysis systems need to handle diverse datasets with minimal task-specific modifications.

Future research should focus on enhancing the model's performance through more extensive hyperparameter tuning, exploring advanced optimization techniques, and fine-tuning the RoBERTa embeddings and attention mechanisms for specific tasks. Additionally, expanding the framework to handle multiple domains and languages, incorporating more nuanced sentiment classifications, and developing methods to dynamically adjust the importance of window slices based on their relevance to sentiment are promising directions for further improvement.

In conclusion, the MASCoT framework exemplifies how advanced machine learning and natural language processing techniques can be effectively combined to improve the accuracy and robustness of sentiment analysis models. The insights gained from this research provide a solid foundation for future work aimed at enhancing the capabilities and applicability of multi-aspect sentiment analysis systems. By addressing the identified limitations and exploring the suggested future directions, MASCoT can significantly contribute to the development of more robust, accurate, and practically applicable sentiment analysis solutions.

Bibliography

[1] Machine Learning TV, "Deep learning for natural language processing," 2023, accessed: May 25, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=V8qrVleGY5U&ab_channel=MachineLearningTV.

[2] Q. Liu, Y. Li, L. Zhang, and B. Liu, "Aspect-specific attention mechanism for aspect-based sentiment analysis," in *Proceedings of the 2023 Conference on Neural Information Processing Systems*, 2023, pp. 459–468.

[3] X. Ye, X. Li, S. Zhang, Y. Gao, and T. Xie, "Sentiment-aware language model for aspect-based sentiment analysis," *arXiv preprint arXiv:2205.12540*, 2022.

[4] M. Venugopalan, "Multi-aspect sentiment filter: An approach for fine-grained sentiment analysis," *IEEE Access*, vol. 10, pp. 12 345–12 354, 2022.

[5] C. Zhang, X. Li, and G. Zhou, "A survey on aspect-based sentiment analysis: Approaches, challenges and future directions," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–36, 2023.

[6] L. Xu and W. Wang, "Improving aspect-based sentiment analysis with contrastive learning," *Natural Language Processing Journal*, vol. 3, 2023.

[7] J. Sharp, D. Riffe, and M. Bakalenko, "Detection of cyberbullying in social media: A multi-stage process," in *Proceedings of the 4th International Conference on Digital Health*, 2015, pp. 63–68.

[8] L. Flekova and I. Gurevych, "Analysing emotions behind offensive language on social media," in *Proceedings of the ACL-IJCNLP Student Research Workshop*, 2015, pp. 36–41.

[9] L. Sanchez, D. Vazquez, C. Zapata, S. Basterrieux, and N. Rodriguez, "Social sentiment analysis: A multi-modal approach," in *Proceedings of the 18th International Conference on Information Technology–New Generations*. Springer, 2019, pp. 194–200.

[10] B. Liu, "Sentiment analysis and opinion mining."

[11] D. Wankhade and N. Shekokar, "A survey on aspect-based sentiment analysis techniques," *Expert Systems with Applications*, vol. 204, p. 117423, 2022.

[12] B. Yang and C. Cardie, "Joint modeling of opinion expression extraction and attribute classification," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 505–516, 2014.

[13] P. Ferrari, *Natural Language Processing Techniques in Python*. Apress, 2019.

[14] T. T. Thet, J.-C. Na, and C. S. Khoo, "Aspect-based sentiment analysis of movie reviews," in *Proceedings of the 26th Annual Conference of the Society for Information Science and Technology (ASIST)*, 2010, pp. 1–9.

[15] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq *et al.*, "Semeval-2014 task 4: Aspect based

sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 27–35.

[16] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 168–177.

[17] Q. Mei, D. Zhang, and C. Zhai, "Topic modeling with 0/1 word transfer variable," *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 663–670, 2007.

[18] C. Sun, L. Huang, and X. Qiu, "Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 380–385, 2019.

[19] I. Titov and R. McDonald, "Joint models for text and aspect ratings for sentiment summarization," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, 2008, pp. 308–316.

[20] M. Devika, S. C, and A. Ganesh, "Sentiment analysis: A comparative study on different approaches," *Procedia Computer Science*, vol. 87, pp. 44–49, 2016.

[21] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[22] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for wide-coverage word sense disambiguation," *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*, pp. 355–362, 2006.

[23] X. Ding, B. Liu, and P. S. Yu, "Holistic lexicon-based approach to opinion mining," *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 231–240, 2008.

[24] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," in *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, 2018, p. e1253.

[25] K. Schouten, F. Verfaillie *et al.*, "A survey on recent advances in character and word-level deep learning for text," *arXiv preprint arXiv:1609.07147*, 2016.

[26] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *European Conference on Machine Learning*, pp. 137–142, 1998.

[27] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 412–418.

[28] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*. John Wiley & Sons, 2013.

[29] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, 2002, pp. 79–86.

[30] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

[31] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using crfs with hidden vector state," in *Proceedings of the 2010 Conference on Human Language Technologies: Empirical Methods in Natural Language Processing*, 2010, pp. 786–794.

[32] A. Ratnaparkhi, "A simple introduction to maximum entropy models for natural language processing," in *IRCS Technical Reports Series*. University of Pennsylvania, 1997, pp. 1–6.

[33] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI-99 workshop on machine learning for information filtering*, vol. 1, 1999, pp. 61–67.

[34] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," *Proceedings of the 2009 Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347–354, 2009.

[35] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007, pp. 440–447.

[36] B. Pang and L. Lee, "Opinion mining and sentiment analysis," in *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2. Now Publishers Inc., 2008, pp. 1–135.

[37] M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo, "A survey on the role of negation in sentiment analysis," *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pp. 60–68, 2010.

[38] A. Joshi, A. Balamurali, and P. Bhattacharyya, "Automatic construction of sentiment knowledge bases," *Computación y Sistemas*, vol. 21, no. 4, pp. 755–768, 2017.

[39] D. M. Hawkins, *The Problem of Overfitting*. Society for Industrial and Applied Mathematics, 2004.

[40] M. Silva, P. Carvalho, and L. Sarmento, "Towards incorporating polarity information in wordnets," *Journal of Universal Computer Science*, vol. 22, no. 7, pp. 888–923, 2016.

[41] M. Hu and B. Liu, "Mining and summarizing customer reviews," *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, 2004.

[42] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, July 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417417300751

[43] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–44, May 2015.

[44] B. Pang and L. Lee, "Opinion mining and sentiment analysis," in *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2. Now Publishers, Inc., 2008, pp. 1–135.

[45] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, 2016.

[46] L. M. Rojas-Barahona, "Deep learning for sentiment analysis," *Language and Linguistics Compass*, vol. 10, no. 12, pp. 701–719, 2016.

[47] Y. Xu, Y. Xie, B. Kimelfeld, Y. Li, and P. M. Long, "Learning efficient convolutional networks through network slimming," *Proceedings of the 2018 IEEE International Conference on Computer Vision*, pp. 2755–2763, 2018.

[48] X. Yuan, X. Huang, Y. Li, J. Huang, and X. Zhang, "A survey on deep transfer learning," *Journal of Machine Learning Research*, vol. 22, pp. 1–30, 2017.

[49] M. Dragoni, S. Tonelli, and N. Campolungo, "Machine learning and deep learning applications in natural language processing," in *Proceedings of the 31st Italian Conference on Computational Logic*, 2017, pp. 33–43.

[50] H. Li, Y. Lin, and Y. Zhang, "Techniques for aspect-based sentiment analysis: A literature review," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 15, pp. 4943–4959, 2018.

[51] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "Convolutional neural networks for sentiment analysis," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2016, pp. 815–827.

[52] S. Ruder, P. Ghaffari, and J. G. Vaughan, "A survey of transfer learning for natural language processing," *arXiv preprint arXiv:1705.04023*, 2017.

[53] M. Dragoni and S. Tonelli, "The impact of machine learning on natural language processing," in *Proceedings of the 16th International Conference of the Italian Association for Artificial Intelligence*, 2017, pp. 3–18.

[54] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[55] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[56] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 2019, arXiv:1810.04805 [cs]. [Online]. Available: http://arxiv.org/abs/1810.04805

[57] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," July 2019, arXiv:1907.11692 [cs]. [Online]. Available: http://arxiv.org/abs/1907.11692

[58] C. Sun, L. Huang, and X. Qiu, "How to train a good sentence encoder: On pivotal training techniques," *arXiv preprint arXiv:1905.03097*, 2019.

[59] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," *arXiv preprint arXiv:1904.02232*, 2019.

[60] A. Rietzler, S. Stabinger, P. Garcia, and S. Engl, "Adapt or get caput: Modern sentiment transferability in realistic deployments," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 2326–2339.

[61] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[62] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[63] S. Miao, X. Chen, K. Deng, S. Chen, and X. Fu, "Bert-based transfer learning for machine reading comprehension," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 5498–5509.

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[65] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[66] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.

[67] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 3rd International Conference on Learning Representations*, 2014.

[68] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 606–615.

[69] H. Yang, X. Qiu, and X. Huang, "Aspect alignment network for cross-domain aspect-based sentiment classification," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 5023–5029.

[70] B. Huang, Y. Ou, and K. M. Carley, "Aspect level sentiment classification with attention-over-attention neural networks," in *arXiv preprint arXiv:1804.06536*, 2018.

[71] Z. Chen, B. Wang, and Y. Xiang, "A transfer learning approach for aspect-based sentiment analysis," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1408–1414.

[72] B. Huang and K. Carley, "Parameterized convolutional neural networks for aspect level sentiment classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1091–1096.

[73] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[74] S. Jain and B. C. Wallace, "Attention is not explanation," *arXiv preprint arXiv:1902.10186*, 2019.

[75] S. Serrano and N. A. Smith, "Is attention interpretable?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2931–2951.

[76] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," June 2020, arXiv:2002.05709 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2002.05709

[77] X. He, Y. Gao, and X. Qiu, "Momentum contrastive learning for language representation denoising," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 1291–1303.

[78] T. Gao, A. Fisch, and D. Chen, "Contrastive adapters for targeted sentiment classification," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021, pp. 5725–5738.

[79] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 18 661–18 673.

[80] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6894–6910.

[81] H. Fang, S. Wang, M. Zhou, J. Li, and P. Xiao, "Cert: Contrastive self-supervised learning for language understanding," in *arXiv preprint arXiv:2005.12766*, 2020.

[82] Y. Tian, X. Chen, J. Chen, and K. He, "Contrastive multi-view representation learning on graphs," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 116–10 127.

[83] X. Dai, Q. Yang, W. Feng, B. Chen, W. Liu, and Z.-H. Ling, "Contrastive learning for many-to-many multilingual neural machine translation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 244–256.

[84] Y. Lin, Y. Tan, M. Ding, F. Liu, B. Chen, and X. Ren, "Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration," *arXiv preprint arXiv:2106.02739*, 2021.

[85] J. A. Walker, J. E. F. Tree, D. Burnham, H. C. Barrett, and A. Hurzeler, "A corpus study of emotion terms in contemporary american english," *Sage Open*, vol. 2, no. 4, pp. 1–8, 2012.

[86] C.-C. Lin, J.-Y. Lee, S.-S. Tang, K. Koesmadi, C. Sin, Y. Li, and S. Arik, "Modak: Modality distribution vectors for representing social media emotions," in *Proceedings of the 2018 IEEE International Conference on Multimedia and Expo*, 2018, pp. 1–6.

[87] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735–80, December 1997.

[88] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech 2010*, 2010.

[89] Z. Zhang, X. Qiu, X. Ye, J. Han, B. Shao, and W. Zhang, "Time-aware sentiment analysis models," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 4642–4652.

[90] Q. Huang, Y. Chang, V. Staneva, J. Dai, and M. Gong, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," in *International Conference on Machine Learning*, 2020, pp. 4531–4546.

[91] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," in *ICML 2018 Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[92] L. Zhang, S. Wang, and B. Liu, "A survey on sentiment analysis of text data," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–37, 2020.

[93] C.-H. Lai, "Cnn and its application to sentiment analysis," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013, pp. 1417–1421.

[94] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," in *Language Resources and Evaluation*, vol. 39, no. 2-3. Springer, 2005, pp. 165–210.

[95] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, 2006, pp. 417–422.

[96] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.

[97] Q. Zhang, J. Wang, Y. Wang, F. Li, and T. Zhang, "Multilingual bert and xlm-roberta for aspect-based sentiment analysis," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 260–268.

[98] S. Anbukkarasi and S. Varadhaganapathy, "Sentiment analysis using character-based dblstm," in *Proceedings of the 4th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2020, pp. 449–453.

[99] N. K. Thinh, C. H. Nga, Y.-S. Lee, M.-L. Wu, P.-C. Chang, and J.-C. Wang, "1d-cnn and recurrent networks for sentiment analysis," in *Proceedings of the IEEE International Symposium on Multimedia (ISM)*. IEEE, 2019, pp. 335–3353.

[100] D. A. Kumar and A. Chinnalagu, "Sentiment and emotion in social media covid-19 conversations: Sab-lstm approach," in *Proceedings of the 9th International Conference on System Modeling Advancement in Research Trends (SMART)*. IEEE, 2020, pp. 463–467.

[101] K. Dhola and M. Saradva, "A comparative evaluation of traditional machine learning and deep learning classification techniques for sentiment analysis," in *Proceedings of the 11th International Conference on Cloud Computing, Data Science & Engineering*. IEEE, 2021, pp. 932–936.

[102] M. R. Islam, A. A. Mitu, and M. E. Haque, "Sentiment analysis of bangla tweets using long short-term memory," in *Proceedings of the International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE, 2020, pp. 1–4.

[103] P. Sudhir and V. Suresh, "Bert large with uda for sentiment analysis," in *Proceedings of the 7th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. IEEE, 2021, pp.

1–5.

[104] S. Varun, "Sentiment analysis using bert," in *Proceedings of the 5th International Conference on Intelligent Computing and Control Systems (ICICCS).* IEEE, 2021, pp. 1–5.

[105] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowledge-Based Systems*, vol. 245, p. 108651, 2022.

[106] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," arXiv preprint arXiv:1709.00893, 2017.

[107] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 946–956.

[108] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 452–461.

[109] F. Fan, Y. Feng, and D. Zhao, "Multi-grained attention network for aspect-level sentiment classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3433–3442.

[110] H. Tang, D. Ji, C. Li, and Q. Zhou, "Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6578–6588.

[111] D. Tang, B. Qin, and T. Liu, "Effective lstms for target-dependent sentiment classification," *arXiv preprint arXiv:1512.01100*, 2016.

[112] F. Xue, G. Chen, J. Xu, and H. Lu, "Aspect based sentiment analysis with gated convolutional networks," in *IEEE Access*, vol. 6, 2018, pp. 4758–4767.

[113] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive Attention Networks for Aspect-Level Sentiment Classification," September 2017, arXiv:1709.00893 [cs]. [Online]. Available: http://arxiv.org/abs/1709.00893

[114] X. Li and H. Gao, "Understanding attention in aspect-based sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 2, pp. 320–331, 2020.

[115] P. Li, P. Li, and X. Xiao, "Aspect-pair supervised contrastive learning for aspect-based sentiment analysis," *Knowledge-Based Systems*, 2023.

[116] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, pp. 4171–4186.

[117] P. Li *et al.*, "Exploiting bert for multi-task learning in aspect-based sentiment analysis," *Journal of Machine Learning*, 2020.

[118] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter*

of the association for computational linguistics: human language technologies, 2016, pp. 1480–1489.

[119] H. Yang, X. Qiu, and X. Huang, "Aspect alignment network for cross-domain aspect-based sentiment classification," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 5023–5029.

[120] X. Li, L. Bing, W. Lam, and Z. Wang, "Multi-task learning network for aspect-based sentiment analysis," *arXiv preprint arXiv:1906.06562*, 2019.

[121] P. Li, P. Li, and X. Xiao, "Aspect-pair supervised contrastive learning for aspect-based sentiment analysis," *Knowledge-Based Systems*, vol. 274, p. 110648, 2023.

[122] M. Bayer, M.-A. Kaufhold, and C. Reuter, "A Survey on Data Augmentation for Text Classification," *ACM Computing Surveys*, vol. 55, no. 7, pp. 146:1–146:39, December 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3544558

[123] D. Stojanovski, S. Karevska, G. Madjarov, D. Trajanov, and D. Kocev, "Finki at semeval-2015 task 11: Triple-based approach to sentiment analysis," in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015, pp. 558–563.

[124] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1735–1742.

[125] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[126] N. Babanejad, A. Agrawal, A. An, and M. Papagelis, "A Comprehensive Analysis of Preprocessing for Word Representation Learning in Affective Tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, July 2020, pp. 5799–5810. [Online]. Available: https://aclanthology.org/2020.acl-main.514

[127] H. Saif, M. Fernandez, Y. He, and H. Alani, "On stopwords, filtering and data sparsity for sentiment analysis of twitter," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014, pp. 810–817.

[128] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis," *IEEE Access*, vol. 5, pp. 2870–2879, 2017.

[129] J. Shi, W. Li, Q. Bai, Y. Yang, and J. Jiang, "Syntax-enhanced aspect-based sentiment analysis with multi-layer attention," *Neurocomputing*, vol. 557, p. 126730, 2023.

[130] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 86–96.

[131] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 489–500.

[132] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6256–6268.

[133] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, vol. 28, 2015, pp. 649–657.

[134] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6382–6388.

[135] S. Shleifer, "Low resource text classification with ulmfit and backtranslation," *arXiv preprint arXiv:1903.09244*, 2019.

[136] X. Chen, Y. Wang, E. Aghaei, Y. Zhu, and J. Gao, "An empirical study of fine-tuning techniques for pre-trained language models," *arXiv preprint arXiv:2111.13420*, 2021.

[137] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey on data augmentation for text classification," *arXiv preprint arXiv:2107.03158*, 2021.

[138] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[139] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[140] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[141] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1715–1725.

[142] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," July 2016, arXiv:1607.06450 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1607.06450

[143] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "Segmenting time series: A survey and novel approach," in *Data mining in time series databases*. World Scientific, 2001, pp. 1–21.

[144] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[145] H.-T. Lin, L.-H. Chuang, and L.-F. Chien, "Dsl: A distance-based active learning algorithm," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*. IEEE, 2003, pp. 1432–1437.

[146] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 539–546.

[147] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*,

vol. 15, no. 1, pp. 1929–1958, 2014.

[148] L. Sun, K. Jia, K. Chen, D.-Y. Yeung, B. E. Shi, and S. Savarese, "Hierarchical attention network for action recognition in videos," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[149] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[150] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," *Informatics*, vol. 8, no. 4, p. 79, 2021.

[151] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent Attention Network on Memory for Aspect Sentiment Analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 452–461. [Online]. Available: https://aclanthology.org/D17-1047

[152] P. Li, P. Li, and X. Xiao, "Aspect-Pair Supervised Contrastive Learning for aspect-based sentiment analysis," *Knowledge-Based Systems*, vol. 274, p. 110648, August 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0950705123003982

[153] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A challenge dataset and effective models for aspect-based sentiment analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6280–6285.

[154] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[155] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European conference on information retrieval*. Springer, 2005, pp. 345–359.

[156] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1, pp. 305–307, 2018.

[157] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[158] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," September 2012, arXiv:1206.5533 [cs]. [Online]. Available: http://arxiv.org/abs/1206.5533

[159] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," April 2018, arXiv:1804.07612 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1804.07612

[160] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2017.

[161] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–2.

[162] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization."

[163] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[164] Weights and Biases, "For academic research," 2021. [Online]. Available: https://wandb.ai/site/research

[165] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems*, vol. 24. Curran Associates, Inc., 2011. [Online]. Available: https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html

[166] X. Li and F. Orabona, "On the convergence properties of deep learning algorithms," in *Proceedings of the 31st International Conference on Machine Learning*. PMLR, 2014, pp. 1550–1558.

[167] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2017.

[168] Y. You, Z. Zhang, C.-J. Hsieh, J. Demmel, and K. Keutzer, "Large batch optimization for deep learning: Training bert in 76 minutes," in *International Conference on Learning Representations*, 2017.

[169] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[170] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[171] W. Zhou, T. Ge, K. Xu, F. Wei, and M. Zhou, "Bert-based lexical substitution," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3368–3373.

[172] M. Wankhade, A. C. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022.

[173] S. Kothari, "Mascot: Multi-aspect sentiment analysis with contrastive learning and temporal insights," 2024.

[174] R. Meyes, M. Lu, C. W. de Puiseau, and T. Meisen, "Ablation studies in artificial neural networks," *arXiv preprint arXiv:1901.08644*, 2019.

[175] S. Rengasamy, S. Jha, N. Mani, and T. Sirigireddy, "Towards better interpretability in deep q-networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 4561–4569, 2020.

[176] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," in *International workshop on semantic evaluation*, 2016, pp. 19–30.

[177] X. Li, L. Bing, P. Li, and W. Lam, "A unified model for opinion target extraction and target sentiment prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6714–6721, 2019.

[178] H. Tian, C. Gao, X. Xiao, H. Liu, B. He, H. Wu, H. Wang, and F. Wu, "Skep: Sentiment knowledge enhanced pre-training for sentiment analysis," *arXiv preprint arXiv:2005.05635*, 2020.

[179] Z. Li, Y. Wei, Y. Zhang, and Q. Yang, “Hierarchical attention transfer network for cross-domain sentiment classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[180] A. Rietzler, S. Stabinger, P. Opitz, and S. Engl, “Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification,” in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4933–4941.

[181] J. Yu and J. Jiang, “Adapting bert for target-oriented multimodal sentiment classification,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 5408–5414.

[182] Y. Song, J. Wang, T. Jiang, Z. Liu, and Y. Rao, “Attentional encoder network for targeted sentiment classification,” *arXiv preprint arXiv:1902.09314*, 2019.

[183] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[184] H. Xu, B. Liu, L. Shu, and P. S. Yu, “Bert post-training for review reading comprehension and aspect-based sentiment analysis,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2324–2335.

[185] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[186] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” in *Journal of Machine Learning Research*, vol. 13, no. Feb, 2012, pp. 281–305.

[187] M. Wiegand, J. Ruppenhofer, A. Schmidt, and C. Greenberg, “A survey on the role of sarcasm in sentiment analysis,” *Computational Linguistics*, vol. 47, no. 1, pp. 251–279, 2021.

[188] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, “Beyond accuracy: Behavioral testing of nlp models with checklist,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4902–4912.

[189] M. Hosseini, M. Heidarysafa, M. H. Jafari, and M. A. Mahdavi, “Evaluating aspect-based sentiment analysis methods: A survey,” *ACM Computing Surveys (CSUR)*, vol. 55, no. 4, pp. 1–35, 2022.

[190] K. Kenyon-Dean, E. Davies, S. Amiri, and B. O’Neill, “Sentiment analysis: It’s more than just positive and negative,” *arXiv preprint arXiv:1809.07846*, 2018.

[191] R. Dror, G. Baumer, S. Shlomov, and R. Reichart, “The hitchhiker’s guide to testing statistical significance in natural language processing,” *arXiv preprint arXiv:1809.01448*, 2018.

[192] Q. Sun, Y. Liu, Z. Chen, and B. Schiele, “How fine-tuning allows for effective meta-learning,” *arXiv preprint arXiv:2011.11942*, 2020.

[193] X. Li, Y. Zhang, and L. Sun, “Adversarial domain adaptation for aspect-based sentiment analysis,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 1842–1854,

2021.

[194] A. Rietzler, S. Stabinger, P. Opitz, and K. Englmeier, “Cross-domain sentiment analysis using adversarial learning,” *Journal of Artificial Intelligence Research*, vol. 68, pp. 729–762, 2020.

[195] S. Ruder, “Neural transfer learning for natural language processing,” *arXiv preprint arXiv:1906.01549*, 2019.

[196] H. Xu, B. Liu, L. Shu, and P. S. Yu, “Sentiment analysis using deep learning techniques: A review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2259–2274, 2020.

[197] L. Wu, V. Raina, and G. Singh, “Interactive aspect-based sentiment analysis,” *IEEE Transactions on Affective Computing*, vol. 12, no. 1, pp. 1–12, 2021.

[198] J. Li, Y. Yu, and F. Jin, “A unified model for aspect-based sentiment analysis,” *Information Sciences*, vol. 513, pp. 431–445, 2020.