

محمد القحطاني

sub_01.docx

 Midocean University

Document Details

Submission ID

trn:oid::2945:406746898

Submission Date

Aug 11, 2026, 8:45 PM GMT+3

Download Date

Aug 12, 2026, 12:01 AM GMT+3

File Name

sub_01.docx

File Size

3.1 MB

53 Pages

11,461 Words

68,227 Characters

29% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely revised by AI.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.



33 AI-generated 29%

Likely generated or likely generated and revised by AI.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely human generated as AI generated and likely AI generated as human generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.



Enhancing Arabic Sentiment Analysis via Domain-Adaptive Transformer Feature Extraction and Machine Learning Classification

Prepared By

Name: Nasser Alshalwan

ID No.: 253000228

Name: Mohammed Alqahtani

ID No.: 253000264

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: Dr. Wessam Ahmed

Collage of Informatics, Midocean University, Comoros

Email:.....

Acknowledgements

Special thanks to Dr. Wessam.

Abstract

Arabic sentiment analysis remains challenging due to linguistic diversity, dialectal variation, informal expressions, and noisy social media text. This thesis proposes an enhanced framework for binary Arabic sentiment classification by leveraging MARBERTv2 contextual representations and task-specific transformer adaptation. Experiments were conducted on the Arabic Sentiment Analysis Dataset (ASAD), comprising 26,785 positive and negative tweets. The study systematically evaluates traditional TF-IDF-based machine learning models, MARBERTv2-based CNN model and a fully fine-tuned MARBERTv2 model. The results demonstrate a clear advantage of transformer adaptation over traditional approaches. While the strongest TF-IDF baseline, SVM, achieved 90.76% accuracy and 90.69% F1-score, the CNN with partially unfrozen MARBERTv2 layers achieved 95.20% accuracy, 96.79% recall, and 95.28% F1-score. The fine-tuned MARBERTv2 achieved the best overall performance, reaching 95.26% accuracy, 94.73% precision, 95.86% recall, and 95.29% F1-score. These findings demonstrate that task-specific adaptation of MARBERTv2 provides powerful contextual representations for Arabic sentiment classification and achieves competitive performance without relying on complex transformer ensembles.

Keywords: Arabic Sentiment Analysis, Natural Language Processing, MARBERTv2, Machine Learning, CNN, TF-IDF, Arabic Tweets.

List of Publications

Arabic Sentiment Analysis: A Survey of Methods, Challenges, and Future Directions

Table of Contents

List of AbbreviationsError! Bookmark not defined.

List of Figures.....Error! Bookmark not defined.

List of TablesError! Bookmark not defined.

Chapter 1 Introduction

1.1 Overview

1.2 Problem Definition

1.3 Thesis Objectives

1.4 Thesis Contribution

1.5 Thesis Organization

Chapter 2 Background and Related Work

2.1 Introduction

2.2 Background

2.2.1 Introduction to Sentiment Analysis

2.2.2 Sentiment Analysis using Machine Learning

2.2.3 Sentiment Analysis using Deep Learning

2.2.4 Sentiment Analysis Using Transformer-Based Models

2.2.5 Performance metrics

2.3 Literature Review

2.4 Comparative Analysis of Related Work

2.5 Chapter Summary

Chapter 3 Proposed Methodology

3.1 Introduction

3.2 Proposed Framework

- 3.3 Dataset Description
- 3.4 Data Preprocessing
- 3.5 Feature Representation
 - 3.5.1 Character-Level TF-IDF for Machine Learning Models
 - 3.5.2 MARBERTv2 Model
 - 3.5.3 MARBERTv2 Token Representations for Neural Models
- 3.6 Machine Learning Models
- 3.7 Deep Learning Models
 - 3.7.1 CNN with Frozen MARBERTv2 Features
 - 3.7.2 CNN with the Last Four MARBERTv2 Layers Unfrozen
- 3.8 Transformer-Based Models
 - 3.8.1 MARBERTv2 Fine-Tuning
- 3.9 Chapter Summary
- Chapter 4 Experimental Results and Discussion
 - 4.1 Introduction
 - 4.2 Experimental Setup
 - 4.3 Results of Machine Learning Models
 - 4.4 Results of Deep Learning Models
 - 4.5 Results of Transformer-Based Models
 - 4.6 Discussion of Findings
 - 4.7 Limitations
 - 4.8 Chapter Summary
- Chapter 5 Conclusion and Future Work
- References

List of Abbreviations

- **Acc.:** Accuracy
- **AraBERT:** Arabic Bidirectional Encoder Representations from Transformers
- **ASAD:** Arabic Sentiment Analysis Dataset
- **ASTD:** Arabic Sentiment Tweets Dataset
- **BiGRU:** Bidirectional Gated Recurrent Unit
- **CAMeLBERT:** Carnegie Mellon University Arabic BERT
- **CNN:** Convolutional Neural Network
- **F1-score:** F1-score
- **FN:** False Negative
- **FP:** False Positive
- **LSTM:** Long Short-Term Memory
- **MARBERT:** Masked Arabic Bidirectional Encoder Representations from Transformers
- **ML:** Machine Learning
- **NLP:** Natural Language Processing
- **Pr.:** Precision
- **R:** Recall
- **SVM:** Support Vector Machine
- **TF-IDF:** Term Frequency–Inverse Document Frequency
- **TN:** True Negative
- **TP:** True Positive

List of Figures

1	Logistic Regression S-shaped curve	6
---	--	---

2	SVM decision boundary with maximum margin	7
3	XGBoost sequential boosting process	8
4	CNN architecture for text classification	9
5	LSTM structure diagram	10
6	BiGRU architectural diagram	10
7	AraBERT feature extraction model architecture (MTL)	11
8	Fine-tuned MARBERT model architecture	12
9	CAMeLBERT architecture	13
10	Proposed Framework	24
11	Distribution of Positive and Negative Tweets Across the Dataset Par.	25
12	Word Clouds of Positive and Negative Tweets After Preprocessing ...	26
13	Training and Validation Learning Curves of the Partially Fine-Tuned.	35
14	Overfitting Analysis of the Partially Fine-Tuned CNN Model	35
15	Confusion Matrix of the CNN with the Last Four MARBERTv2 Lay.	36
16	Performance Comparison of All Evaluated Models	38

List of Tables

1	Comparison of sentiment analysis approaches in Arabic NLP	13
---	---	----

2	Comparative Study of Related Work	20-21
3	Distribution of the Dataset Across the Training, Validation, and T. ..	25
4	Examples of Arabic Tweets Before and After Preprocessing	27
5	TF-IDF Configuration	28
6	MARBERTv2 Representation Configuration	28
7	Architecture of the CNN with Frozen MARBERTv2 Feature	29
8	Architecture of the CNN with the Last Four MARBERTv2 Lay.	30
9	Performance of Machine Learning Models on the ASAD Test Set ...	31
10	Performance of Deep Learning Models on the ASAD Test Set	34
11	Distribution of Positive and Negative Tweets Across the Dataset P.	34
12	Performance of the Transformer-Based Model on the ASAD Test ...	36

Chapter 1 Introduction

1.1 Overview

Sentiment analysis is widely used in natural language processing because it helps identify the opinions, emotions, and attitudes expressed in text. Its importance has increased with the rapid growth of online communication, especially on social media platforms, review websites, and other digital spaces where users frequently share their views. In Arabic, sentiment analysis has received growing attention in recent years. However, recent survey studies still show that Arabic sentiment analysis is less mature than sentiment analysis in high-resource languages, particularly in modern deep learning settings [1]. One of the main challenges in Arabic sentiment analysis is the nature of the Arabic language itself. Arabic is used in different forms, including Modern Standard Arabic and many regional dialects, and these forms often differ in vocabulary, spelling, and style. This makes sentiment classification more difficult, especially when the data comes from informal environments such as social media. However, the challenge is bigger when sentiment data comes from different domains like tweets, reviews, or online discussions. Patterns from one domain may not work well in another. Also, recent studies showed that cross-domain and cross-dialect sentiment analysis is still difficult and needs better methods to deal with the variation in Arabic contexts [2]. Also, progress in this field is not only about making better models. It is also about using good and updated benchmarks to test them. Some recent studies showed that some Arabic sentiment benchmarks are still limited and do not always show the real power of new models. So, benchmark quality is still important for improving Arabic sentiment analysis research [3]. At the same time, recent research suggests that hybrid approaches are still promising for Arabic sentiment classification. Combining transformer-based contextual representations with traditional text features may be useful because sentiment in Arabic can appear through both contextual meaning and surface-level lexical patterns. Some recent studies on Arabic dialect sentiment classification also support this direction by showing the value of integrating transformer representations with traditional features such as TF-IDF [4]. Also, Arabic sentiment analysis is now focusing more on domain variation, benchmark quality, and combining different features. So, it

makes sense to look at a framework that uses domain-adaptive transformer feature extraction and machine learning classification in this study.

1.2 Problem Definition

Arabic sentiment analysis still has many challenges because Arabic texts are different in dialects, domains, and writing styles. Also, this makes sentiment classification difficult, especially when the data is not the same as the training data. Also, working well on one model does not mean all results will be good. It can give poor results when the text is noisy, informal, or from special domain. Also, the problem is more complex in social media data, where there are many dialects, non-standard spelling, and informal words.

1.3 Thesis Objectives

This study's primary objective is to explore some more effective method for the task of Arabic sentiment analysis. For this purpose, it employs domain-adaptive transformer features and machine learning for the classification task. Additionally, it aims at enhancing the quality of sentiment classification for Arabic language texts, especially when these texts are from divergent domains and writing styles. Meanwhile, it pertains to how the combination of transformer features with classification could be more effective. Moreover, it aims at constructing a more enhanced and balanced method for the task of Arabic sentiment analysis particularly in more complex context.

1.4 Thesis Contribution

This study is expected to contribute to Arabic sentiment analysis in the following ways:

- It proposes a framework that combines domain-adaptive transformer-based feature extraction with machine learning classification. Specifically, MARBERTv2 is

employed to extract contextual sentiment-aware features from Arabic text, which are then passed to conventional machine learning classifiers for sentiment prediction.

- It evaluates the proposed framework on the ASAD dataset, assessing its effectiveness in classifying the sentiment of Arabic tweets under real-world, noisy social media conditions.
- The study establishes TF-IDF-based Logistic Regression and Support Vector Machine (SVM) models as baseline approaches and compares their performance with MARBERTv2-based models.
- The experimental results examine the effect of different representation and adaptation strategies, including fixed transformer representations, CNN-based contextual representations, partial transformer-layer fine-tuning, and full MARBERTv2 fine-tuning.

1.5 Thesis Organization

The remainder of the thesis is structured as follows:

Chapter 2: provides background and related work about Arabic tweet sentiment analysis. Also, it talks about machine learning, deep learning, and transformer-based methods. In addition, it shows some recent studies about Arabic sentiment analysis.

Chapter 3: outlines the proposed approach of the study, including the dataset, preprocessing steps, feature extraction process, classification models, and evaluation metrics.

Chapter 4: presents the experimental results and discussion of the proposed framework, including the performance of the models and the comparison between the proposed approach and the baseline methods.

Chapter 5: concludes the study by summarizing the main findings, highlighting the limitations, and presenting directions for future work.

Chapter 2 Background and Related Work

2.1 Introduction

This chapter provides the background for Arabic tweet sentiment classification. It begins by explaining sentiment analysis and the main challenges that appear in Arabic sentiment analysis, especially in social media and dialectal contexts. It then reviews the main approaches used in sentiment classification, including machine learning, deep learning, and transformer-based models. It also presents the evaluation metrics used in this study and discusses recent related work to clarify the research gap behind the proposed framework.

2.2 Background

2.2.1 Introduction to Sentiment Analysis

Sentiment analysis, or opinion mining, is a common task in Natural Language Processing (NLP). It is mainly used to identify the sentiment expressed in text, such as whether the text is positive, negative, or neutral. In some studies, more detailed sentiment classes are also used depending on the dataset and the aim of the research. So, these days have become more important because people share a lot of text every day on social media, review sites, forums, and other places online. So, sentiment analysis became more important. Moreover, it is also beneficial for many aspects such understanding customers' comments, following the public opinions, checking products and watching brands. It definitely plays a significant role in the social media, whereby the people are giving their views directly and very transparently. Yet, it is not easy to conducting a sentiment analysis; as the meaning of the utterance could be changeable due to the context, sarcasm, negation, ambiguity or single word. But for Arabic language, it is more complex because of the existence of Modern Standard Arabic and many dialects that are differing by the word choice, spellings and style; whereas on the social media, it will be more extreme as the users usually confuse the dialects, encrypt the words by using repetitive alphabet, EMOJIs or others. Also, sentiment analysis can be done at document-level, sentence-level, or aspect-level. In tweet

classification, short-text sentiment analysis is particularly important because each tweet is usually treated as one unit for prediction. Arabic sentiment analysis may also use binary, three-class, or multi-class settings depending on the dataset and the purpose of the study. In this study, the task is formulated as binary sentiment classification. Neutral tweets are excluded, and the remaining tweets are classified as positive or negative. This gives a more focused way to classify Arabic tweet sentiment. Over time, sentiment analysis methods have moved from traditional feature-based approaches to machine learning, deep learning, and transformer-based models. The following sections review these approaches and explain their relevance to Arabic sentiment analysis [5][6].

2.2.2 Sentiment Analysis using Machine Learning

Machine learning is used in sentiment classification by training models on labeled data. Also, the text is converted into features, and TF-IDF is often used in this step. After that, machine learning models like Logistic Regression, SVM, and XGBoost are used for sentiment classification [7].

2.2.2.1 Logistic Regression

Logistic Regression is a common machine learning model for sentiment analysis. Also, before using it, the text data is changed to numbers, for example by using TF-IDF. Then, the model is trained on labeled data to find patterns and predict the sentiment. Researchers usually include Logistic Regression because it is straightforward and gives reasonable results, especially as a baseline model, as shown in Figure 1. Even so, it mainly depends on the quality of the extracted features, and this makes it less capable of understanding contextual meaning compared with newer models [8].

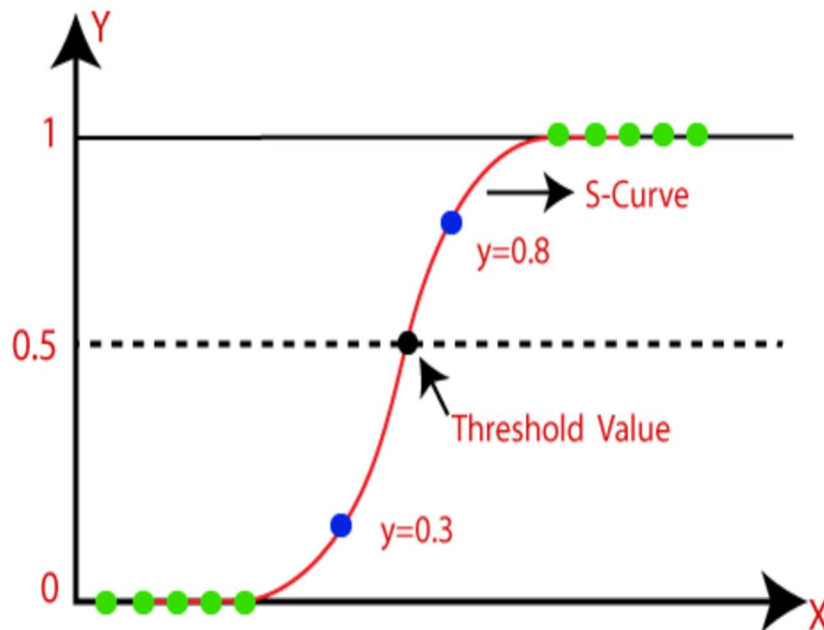


Figure 1. Sample Logistic Regression Function S-shaped curve [9].

2.2.2.2 Support Vector Machine

Support Vector Machine is a well-known machine learning model in sentiment analysis. It is commonly applied after converting text into numerical form using feature extraction methods such as TF-IDF. The main idea of this model is to classify the text by identifying a separating boundary between different sentiment categories. Many studies use Support Vector Machine because it is effective in handling text classification problems and can produce good performance in many cases, as shown in Figure 2. However, its results are still influenced by the quality of the extracted features, which means it may not represent contextual meaning as effectively as more recent deep learning and transformer-based models [10].

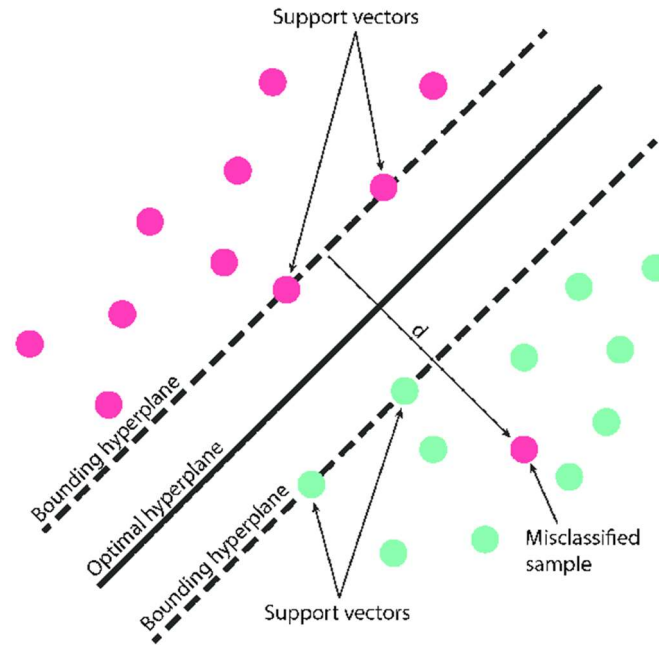


Figure 2. Support Vector Machine (SVM) decision boundary with maximum margin [11].

2.2.2.3 XGBoost

XGBoost is also used in sentiment analysis as a feature-based machine learning model. In most cases, the text is not used directly, but is first changed into numerical representation using methods such as TF-IDF. These features are then used to train the model so it can distinguish between sentiment classes. Also its included in some sentiment analysis studies cause it applies a tree-based boosting strategy, so that can be improve classification performance in feature-driven tasks, as shown in Figure 3. However, its performance still depends on the numerical representation of the text, which makes it less effective in understanding context when compared with deep learning and transformer-based approaches [12].

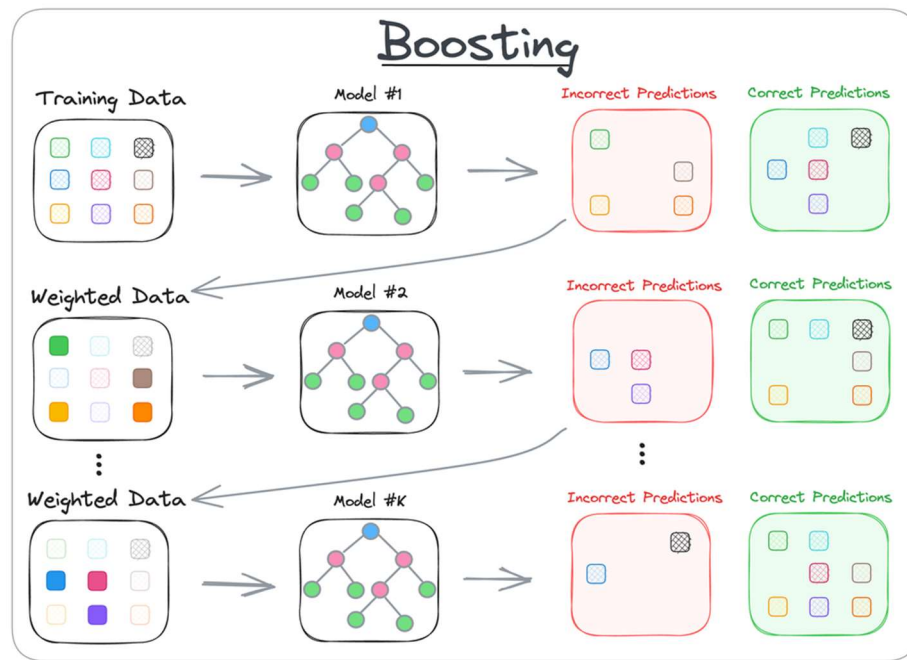


Figure 3. XGBoost sequential boosting process [13].

However, machine learning models are still important in sentiment analysis because they provide clear baseline results. Their effectiveness is closely related to the preprocessing stage and the quality of feature extraction. Also, their ability to represent contextual meaning is more limited than that of newer techniques. Also, in Arabic sentiment analysis, this problem is more difficult. There are many dialects and complex morphology. Also, the informal style is common, especially in tweets [14].

2.2.3 Sentiment Analysis using Deep Learning

Deep learning is a branch of machine learning used in sentiment classification. These models do not depend on feature extraction methods such as TF-IDF as in machine learning (ML), because they can learn text representation automatically from the input data. On the other hand, models such as CNN, LSTM, and BiGRU are often used to classify text based on the patterns learned during training [15].

2.2.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Network is also used in sentiment analysis as a deep learning model. In text classification, it learns useful patterns from the input text and uses them in the classification process. CNN is often applied because it can detect important parts of the text that support the sentiment decision, as shown in Figure 4. However, it may be less effective when the text requires understanding longer word relations compared with some other deep learning models [16].

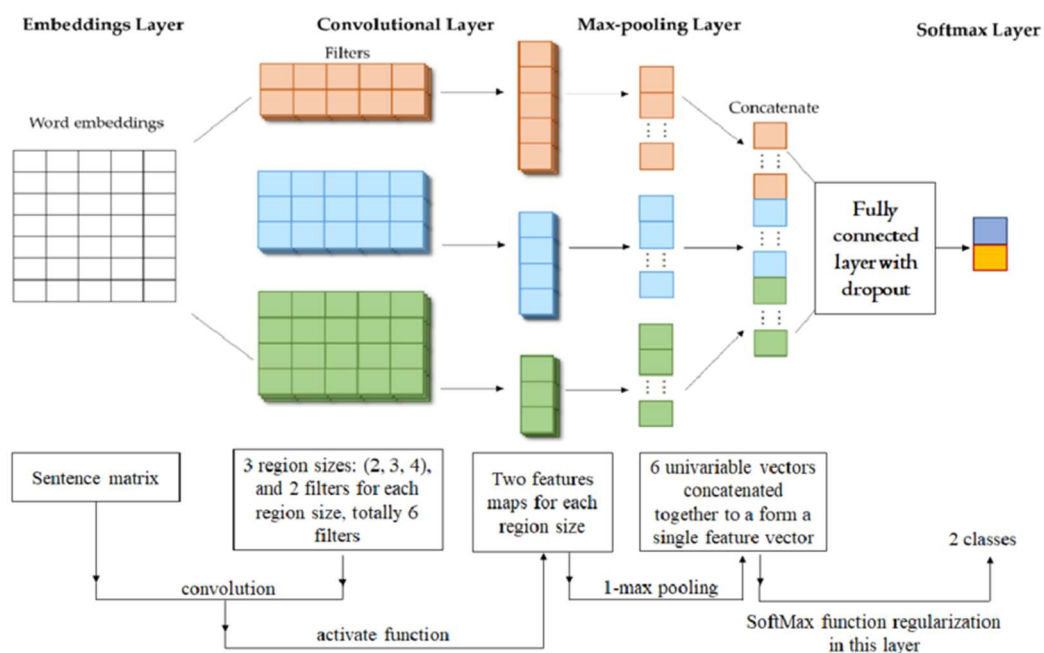


Figure 4. The architecture of a CNN for text classification [17].

2.2.3.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory is used in sentiment analysis to classify sequential text data. Instead of treating words independently, it follows the order of the sequence during learning. This allows the model to make use of information that appears across the text when predicting the sentiment class, as shown in Figure 5. However, it often requires more training effort than simpler models [18].

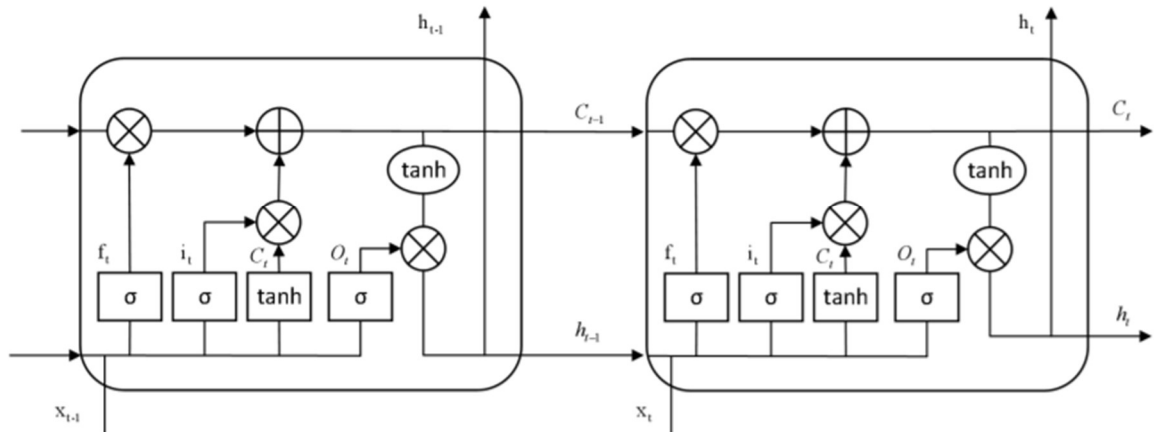


Figure 5. LSTM structure diagram [19].

2.2.3.3 Bidirectional Gated Recurrent Unit (BiGRU)

Bidirectional Gated Recurrent Unit is also applied in sentiment analysis as a deep learning model. It reads the sequence from both directions, which gives the model access to information from the beginning and the end of the text. This can support sentiment classification by providing a better view of the sequence, as shown in Figure 6. However, the model result is still affected by the training data and parameter settings [20].

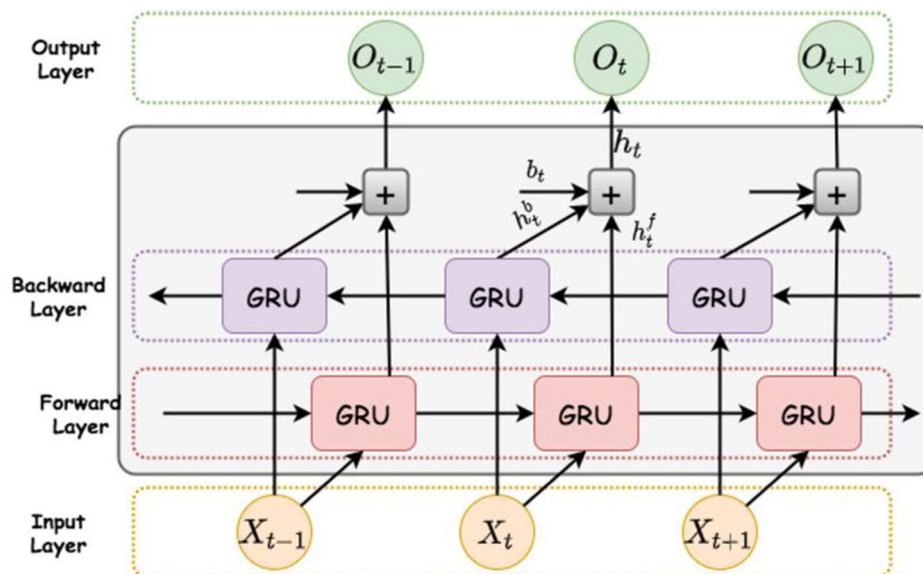


Figure 6. Architectural diagram of the BiGRU model [21].

Deep learning models are widely used in sentiment analysis. CNN, LSTM, and BiGRU are common examples in text classification. At the same time, their results still depend on the quality of the data, and how the model is designed, especially for Arabic text [22].

2.2.4 Sentiment Analysis Using Transformer-Based Models

Transformer-based models are used in sentiment analysis. These models help represent the meaning of text based on context. Also, they have a lot of advantages, one of them is the ability to understand the same word differently depending on the surrounding words. However, in Arabic sentiment classification, it is common to use models such as AraBERT, MARBERT, and CAMeLBERT [23].

2.2.4.1 AraBERT

AraBERT is a transformer-based model introduced for Arabic language understanding. It was pre-trained on large Arabic text corpora and presented as a model can support different Arabic NLP tasks. Cause of this, it has been used in tasks such as sentiment analysis and text classification, as shown in Figure 7 [24].

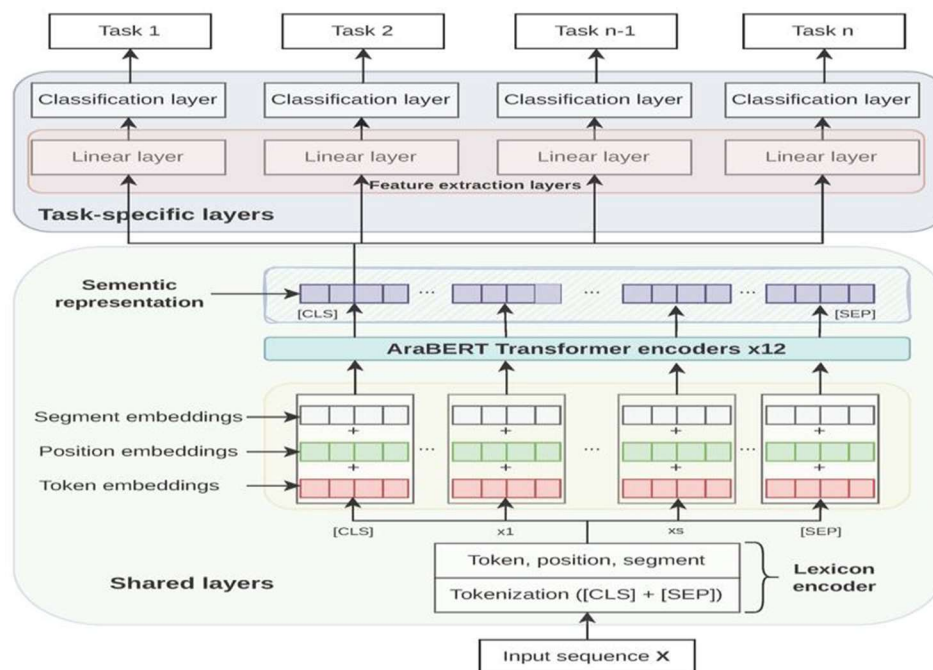


Figure 7. The feature extraction model architecture based on MTL and AraBERT [25].

2.2.4.2 MARBERT

MARBERT is an Arabic transformer-based model developed with a stronger focus on dialectal Arabic and text from social media. It was trained on a large amount of Arabic data, including informal content, cause of this its become a useful for tasks that involve noisy and dialectal text. For this reason, it has been used in Arabic sentiment classification, as shown in Figure 8 [26].

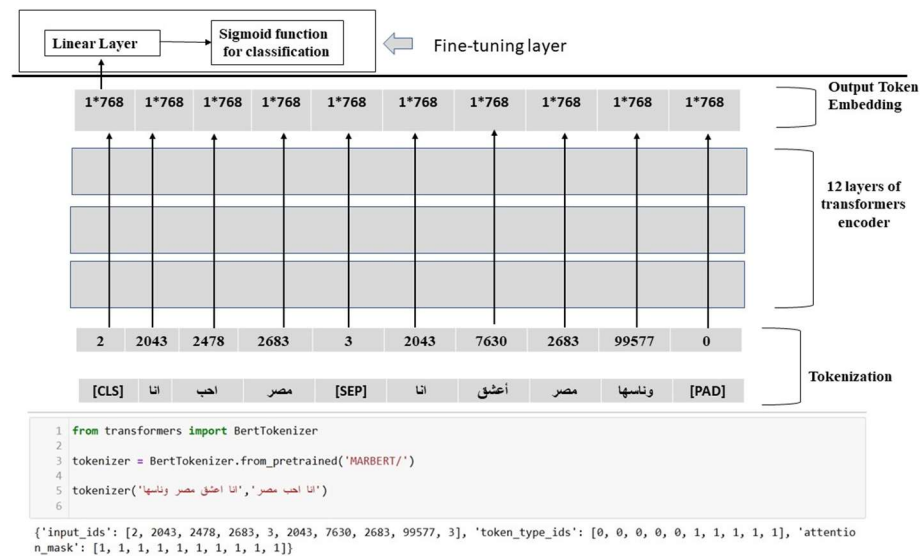


Figure 8. The fine-tuned MARBERT model architecture [27].

2.2.4.3 CAMELBERT

CAMELBERT is one of the Arabic pre-trained language models introduced for Arabic NLP tasks. The related paper focused on the effect of language variant, model size, and task type on model performance. Based on this, CAMELBERT was presented as a model that can support Arabic text classification under different Arabic forms, as shown in Figure 9 [28].

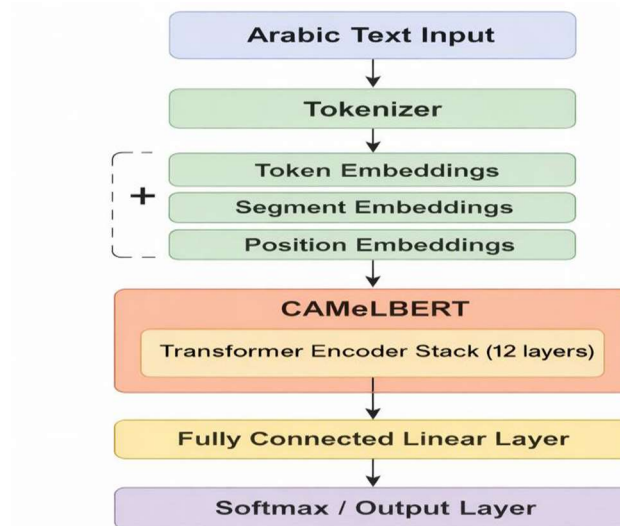


Figure 9. CAMELBERT architecture.

Transformer-based models have shown strong results in recent Arabic sentiment analysis studies. So, this is especially clear in Arabic models that were applied to sentiment tasks and dialectal or noisy text. Although these models usually need more computation and careful fine-tuning, they are still among the most suitable choices for Arabic sentiment classification. Cause of this, they are an important part of the proposed framework [29]. As shown in Table 1, compares the main sentiment analysis approaches used in Arabic NLP.

Table 1. Comparison of sentiment analysis approaches in Arabic NLP.

Approach	Common Models	Suitable Data	Main Advantage	Main Limitation
Machine Learning	Logistic ,SVM ,Regression XGBoost	Structured or relatively clean text represented using features such as TF-IDF	,Strong baseline models ,interpretable ,efficient and useful for comparison	Limited ability to capture semantic ,word order ,relations and contextual meaning
Deep Learning	,LSTM ,CNN BiGRU	Sequential text with semantic ,patterns including noisy text with word dependencies	Better representation learning than traditional machine learning and improved ability to capture local and sequential patterns	Requires more more ,training data and ,tuning effort higher computational cost
Transformer-Based Models	,AraBERT ,MARBERT CAMELBERT	,Arabic tweets ,dialectal text and noisy social media data	Strong contextual modeling and better suitability for complex Arabic sentiment analysis tasks	Requires more computational resources and careful fine-tuning

2.2.5 Performance metrics

To evaluate the performance of the sentiment classification models used in this study, four standard performance metrics were used: accuracy, precision, recall, and F1-score. These metrics are commonly used in classification tasks because they help assess model performance clearly. In classification tasks, the evaluation is usually based on four possible outcomes. True Positives (TP) are the cases that are correctly classified as positive, while True Negatives (TN) are the cases that are correctly classified as negative. False Positives (FP) are the cases that are incorrectly classified as positive, whereas False Negatives (FN) are the cases that are incorrectly classified as negative. Accuracy measures the proportion of correctly classified instances out of the total number of instances. It gives a general indication of the model's overall performance. The accuracy is calculated using Eq. (1) [30].

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (1)$$

Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It shows how accurate the positive predictions are. The precision is calculated using Eq. (2) [31].

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (2)$$

Recall measures the proportion of actual positive instances that are correctly identified by the model. It shows how well the model can detect positive cases. The recall is calculated using Eq. (3) [31].

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (3)$$

F1-score combines precision and recall into a single value by taking their harmonic mean. It is useful when both precision and recall need to be considered together. The F1-score is calculated using Eq. (4) [31].

$$F1\text{-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

2.3 Literature Review

This section introduces recent related studies in Arabic sentiment analysis and focuses on Arabic tweets, dialectal content, noisy user-generated text, and transformer-based sentiment classification. The purpose of this review is to identify the main research directions in recent literature, underline the strengths and limitations of prior work, and clarify the academic position of the proposed study. The reviewed studies also help establish the research limitations that motivate the use of a machine learning approach with a transformer-based framework on the ASAD dataset.

Abdul-Mageed et al. [32] presented the NADI 2022 shared task for dialect identification and dialectal sentiment analysis on Arabic tweets. The study did not focus on a single standalone model but rather established a benchmarking framework to evaluate different systems on nuanced Arabic tweet data. The results showed that the winning system achieved an F1 score of 75.16 in the sentiment subtask. This study is important because it provides a clear benchmark for Arabic tweet sentiment analysis and demonstrates the performance level that strong systems achieved in a competitive setting. However, as a shared-task paper, its main contribution is evaluation and comparison rather than proposing a reusable sentiment framework.

Abdel-Salam et al. [33] developed a system for dialect and sentiment identification in nuanced Arabic tweets using an ensemble of prompt-based, fine-tuned, and multitask BERT models on the NADI 2022 dataset. Their system got first place in the sentiment subtask with F1-PN score 75.15. Also, it showed that using different BERT-based methods can work well for Arabic tweet sentiment analysis, especially when there are small dialect differences. However, the system was made mainly for a shared-task, so it is not easy to use in normal research work.

Bayrak and Issifu [34] explored domain-adapted BERT-based models for Arabic dialect identification and sentiment analysis on tweets. The authors used MARBERT and other Arabic BERT models with special preprocessing for Arabic on NADI 2022 tweets. The best sentiment model got 74.29 Macro F1-PN. Also, it shows that domain adaptation and

preprocessing are important when working with Arabic tweets that have many dialects. And, the performance was still lower than the best recent ensemble systems. And that means domain adaptation alone may not be enough to get the best results.

Qaddoumi [35] presented an ensemble system for Arabic tweet sentiment analysis. The system was a simple averaging of the Arabic BERT-based models provided on NADI 2022 Subtask 2. They placed third place, with 72.49 Macro-PN-F1 in the development set, and 69.86 F1-PN in the test set. After present their system, it can be seen that a simple ensemble can be strong for Arabic tweet sentiment analysis. However, this system performance was still inferior to the top system in the same benchmark.

Belbachir et al. [36] proposed a sentiment analysis system for Algerian Arabic tweets. It used MARBERT with lexical and emoji-based filtering on the SemEval-2023 Task 12 dataset. Also, the filtering step improved the F1 score from 0.732 to 0.736. However, the improvement was small. On other hand, it shows that targeted preprocessing can help sentiment classification in dialect Arabic. Also, the study is important because it works on Algerian Arabic, which is not common in Arabic sentiment analysis. However, the main limitation is that the results are only for one dialect, so it is hard to use them for all Arabic tweets.

Alosaimi et al. [37] proposed a hybrid model for Arabic sentiment analysis by combining AraBERT with LSTM. Also, they tested this model on four Arabic sentiment benchmark datasets. And for improving classification performance, they need to use a transformer and sequential learning as the main goal. Also, the model achieved accuracy above 97%, which shows good potential for combined models in Arabic sentiment analysis. However, the experiments used only benchmark datasets, not Arabic tweets, so maybe the results do not apply directly to noisy or dialectal social media text.

Baniata and Kang [38] developed a Switch Transformer model for Arabic dialect sentiment analysis with Mixture of Experts. The model was tested on Arabic dialect datasets with three-polarity and five-point settings. The result showed 84.02% accuracy. Also, it showed that advanced transformer models can help with sentiment classification in Arabic, cause of dialects and non-standard writing make it difficult. However, the model needs more computation, so on the other hand, it may not be practical for lightweight academic use.

Almaqtari et al. [39] developed a hybrid model for Arabic sentiment analysis using AraBERT, multi-channel CNN, and BiGRU on three Arabic review datasets. The accuracy

reached 96.92% for ACRD, 96.68% for HARD, and 92.93% for LABR. The study also reported that the advantage of combining contextual, local, and sequential features can improve classification performance. Meanwhile, the addressed sentiment analysis of reviews rather than tweets.

Alawneh et al. [40] examined the effect of preprocessing and emoji encoding on Arabic sentiment analysis by using a CNN-LSTM framework on the Arabic Sentiment Twitter Corpus (ASTC). And the best result was 91.85% accuracy. This was after several preprocessing steps. In addition, the steps were to keep non-Arabic words, remove punctuation, use Snowball stemming, change emojis to Arabic text, and use Keras embeddings. Meanwhile, the main importance is that tweet preparation is very important for Arabic sentiment classification, especially in social media, where people use informal language and emojis. However, the main limitation is that the experiments were only done on one tweet dataset.

Mansour et al. [41] proposed a transformer-based ensemble model for dialectal Arabic sentiment classification. Also, the study compared TF-IDF baselines, machine learning models, deep learning models, and transformer models like CAMELBERT, MARBERT, and XLM-RoBERTa on ASTD, ASAD, and TEAD datasets. The ensemble yielded some of the highest performances, with 90.4% mean accuracy, 87.7% F1 and 95% F1 on ASAD. As a result, it is presently one of the reference systems for Arabic tweet sentiment analysis. Due to that, the best system was still an ensemble, and in consequence it is more complex and less easy to interpret than a single system.

Aljomah et al. [42] explored Arabic sentiment analysis on noisy text by using a pre-trained transformer model. Also, the authors fine-tuned CAMELBERT and compared it with AraBERTv02-twitter and MARBERT on the MARSA dataset. Fine-tuned CAMELBERT got 92% accuracy and 91.5% F1-score. This is important because noisy Arabic tweets are hard to classify because of spelling variation, informal words, and dialects. However, the experiments were only on one noisy-text dataset. Also, the study showed that fine-tuning transformers is useful for noisy Arabic social media sentiment tasks.

Jaber et al. [43] developed an ensemble machine learning approach for Arabic sentiment analysis using pre-trained embeddings and ensemble voting on ArTwitter and Syria_Tweets. The results showed strong performance, including 92.43% accuracy and 92.00% F1-score on ArTwitter. This study is useful because it shows that traditional

ensemble machine learning can still achieve high results when supported by strong embeddings. It also reminds us that strong Arabic sentiment performance does not always require a fully end-to-end transformer pipeline. Nevertheless, the method remains more classical in design than recent transformer-based sentiment frameworks.

Alharbi et al. [44] presented the AHaSiS shared task on sentiment analysis for Arabic dialects using a multi-dialect hospitality review dataset. One of the top-performing system achieved a Micro-F1 score of 0.81. So, the study a useful reference for dialect-sensitive evaluation in Arabic sentiment analysis. But, still cause the dataset is based on hospitality reviews rather than tweets, its relevance to tweet sentiment classification remains indirect. Alghamdi et al. [45] examined sentiment analysis in Saudi-English code-switching text and introduced the Saudi-English Sentiment Analysis Code-Switching Dataset (SESA-CSD). Several small language models were evaluated, and the best result came from XLM-RoBERTa, which achieved 95.50% accuracy and 95.53% Macro-F1. These results are strong and recent, and they are also connected to Saudi social media language use. However, the study focuses on code-switching rather than fully Arabic-only tweets, so it is closer to a supportive reference than a directly matching one.

Alsahafi et al. [46] performed sentiment analysis using deep learning methods on Arabic tweets during Riyadh Season. For this, the authors created a dataset out of 33,300 tweets from X and compared CNN, LSTM, CNN-BiLSTM, and BiLSTM with attention with AraBERT embeddings. The best performance was obtained by BiLSTM with attention with AraBERT embeddings, with an accuracy of 83.37%. This is interesting as a more recent work, it concerned tweets and it was linked to the Saudi social media context. However, it is hard to generalize the results to other Arabic tweet dataset due to the fact that the dataset is linked to one event domain only.

2.4 Comparative Analysis of Related Work

From the literature reviewed, it appears that Arabic sentiment analysis has advanced quite far in recent years, and these studies focus much on shared tasks for benchmark-oriented research, such as competitions designed around analyzing a corpora of Arabic Tweets (and other micro-blogs and SMS) that tend to be found in dialect-rich environments and for this specific task. In addition, these experiments are useful in providing baselines in terms of performance, and are a reminder that sentiment classification is challenging in the domain

of Arabic social media. However, some of the experiments are designed to be used in challenge settings and as such their systems may not be easily used as straightforward research application systems. However, much of the work on Tweets and dialectal text is very relevant to this work project while in addition, some experiments relied on Arabic Tweets or noisy social media texts. That showed in better result because of preprocessing, dealing with emojis, and the use of dialects. Regarding to best result in sentiment analysis of Arabic tweets, it should not be only depending on the model, but also depend on dataset and on text preparation before classification. This is relevant in this work because dataset ASAD contained social media content. Thus, use of informal language and dialects is predictable. Also, there is switch from classical ML to deep learning, and now to transformer-based models. Despite the fact that classical ML still can be used as a baseline, particularly through the use of tf-idf features. There are greater numbers of studies, using neural networks based and transformer-based models. To reach better understanding. On the other hand, hybrid deep learning models also performed well, particularly those that employ contextual embeddings (such as in Trans and CapBERT) using an LSTM or a bi-GRU. Moreover, many high-performing deep learning models were tested on general benchmark or review datasets, and not specifically on Twitter data; thus, a direct comparison for classifying sentiment of Arabic social media content becomes difficult. Furthermore, models that base their predictions using transformer architecture are the state of the art for recent classifying sentiment in Arabic. Studies with the MARBERT, CAMeLBERT, XLM-RoBERTa, and similar transformers outperform traditional approaches, particularly where the input contained dialect or noise in the Arabic language. In the work of Mansour et al., machine learning, deep learning, and transformer models were compared across the ASTD, ASAD, and TEAD datasets. Their best ensemble model achieved 95.6% accuracy, 95% precision, 95% recall, and a 95% F1-score on the ASAD dataset. Their best classifier was an ensemble system. Therefore, this approach is more complicated and difficult to understand than applying one single model. Moreover, a solution still needs to be broken down. The authors of most recent state-of-art research work combine ensemble system, treat them as benchmark comparisons, or try using different datasets which are not exactly same as the conducted task. Hence, the proposed project departs to the alternative framework through the application of domain-adaptive transformer feature extraction with machine-learning classifier to the ASAD dataset. The

project applied TF-IDF with Logistic Regression and SVM as baseline models, while MARBERTv2 was used as the main transformer model. The working and methods of the studies discussed above are summarized in Table 2, Comparative Study of Related Work.

Table 2. Comparative Study of Related Work

Authors	Year	Objective	/ Dataset Domain	Model / Technique	Key Result	Limitation
Abdul-Mageed et al.	2022	To benchmark dialect identification and dialectal sentiment analysis on Arabic tweets	/ NADI 2022 Arabic tweets	Shared-task benchmark systems	Winning system achieved 75.16 F1 in sentiment subtask	Benchmark not a ,paper single standalone model
Abdel-Salam et al.	2022	To identify dialect and sentiment in nuanced Arabic tweets	/ NADI 2022 Arabic tweets	fine- ,Prompt-based and ,tuned multitask BERT ensemble	Achieved 1st place with 75.15 F1-PN	Designed mainly for shared-task setting
Bayrak and Issifu	2022	To improve nuanced Arabic dialect and sentiment identification through domain adaptation	/ NADI 2022 Arabic tweets	Domain-adapted MARBERT and Arabic BERT models	Best sentiment model achieved 74.29 Macro F1-PN	Lower than stronger recent ensemble systems
Qaddoumi	2022	To build an ensemble for Arabic tweet sentiment analysis	/ NADI 2022 Arabic dialect tweets	Average ensemble of Arabic BERT models	Macro- 72.49 PN-F1 on 69.86 ,dev F1-PN on ranked ,test 3rd	Simpler ensemble performed below top systems
Belbachir et al.	2023	To improve sentiment analysis of Algerian Arabic tweets	SemEval-2023 Task 12 Algerian / Arabic tweets	MARBERT with lexical and emoji-based filtering	Improved from 0.732 F1 to 0.736 F1	Focused on one dialect only
Alosaimi et al.	2024	To improve Arabic sentiment analysis using a hybrid model	Benchmark Arabic sentiment datasets	LSTM + AraBERT	Reported accuracy above 97%	General benchmark not ,setting tweet-specific
Baniata and Kang	2024	To address Arabic dialect sentiment analysis using advanced transformers	Arabic dialect sentiment datasets	Switch Transformer with Mixture of Experts	Achieved 84.02% accuracy in one setting	High computational cost
Almaqtari et al.	2024	To improve Arabic sentiment classification in consumer reviews	,ACRD ,HARD / LABR Arabic reviews	Multi- + AraBERT + channel CNN BiGRU	96.92% on ,ACRD 96.68% on ,HARD 92.93% on LABR	Review-not ,based tweet-based
Alawneh et al.	2024	To study the effect of preprocessing and emoji encoding on	/ ASTC Arabic Twitter corpus	CNN-LSTM with preprocessing and emoji encoding	Best setup achieved 91.85% accuracy	Evaluated on one tweet dataset only

		Arabic tweet sentiment analysis				
Mansour et al.	2025	To improve dialectal Arabic sentiment classification with transformer ensembles	,ASTD ,ASAD / TEAD Arabic tweets	Ensemble of ,CAMELBER XLM- ,MARBERT RoBERTa and others	Accuracy 95.6% Precision Recall 95% F1- 95% score 95%	Ensemble is heavier and less interpretable
Aljomah et al.	2025	To classify noisy Arabic sentiment text using transformers	/ MARSA noisy Arabic text	Fine-tuned CAMELBER vs AraBERTv02- twitter and MARBERT	92% ,accuracy 91.5% F1- score	One noisy-text dataset only
Jaber et al.	2025	To improve Arabic sentiment analysis using embeddings and ensemble ML	,ArTwitter Syria_Tweets	Pre-trained + embeddings ensemble voting	92.43% ,accuracy 92.00% F1- score on ArTwitter	More classical than recent transformer frameworks
Alharbi et al.	2025	To evaluate sentiment analysis for Arabic dialects in hospitality domain	Multi-dialect hospitality reviews	Shared-task systems	Top system achieved 0.81 Micro-F1	Domain is not ,reviews tweets
Alghamdi et al.	2026	To analyze Saudi-English code-switching sentiment and introduce a new dataset	/ SESA-CSD Saudi-English code-switching social media text	Small language best with ,models XLM-RoBERTa	95.50% ,accuracy 95.53% ,Macro-F1 with 3%– 11% improvement over baseline	Code-switching not ,focus fully Arabic-only tweets
Alsahafi et al.	2026	To analyze sentiment of Arabic tweets during Riyadh Season	tweets 33,300 X (Twitter) /	,LSTM ,CNN ,CNN-BiLSTM + BiLSTM attention with AraBERT embeddings	Best model achieved 83.37% accuracy	Domain-specific to Riyadh Season

2.5 Chapter Summary

This chapter presented the main background of Arabic sentiment analysis, particularly on social media text and dialectal Arabic. It first described the term sentiment analysis, the challenges faced by researchers, the importance of sentiment analysis, and the settings used for sentiment classification task in previous research. Also, the traditional machine learning approaches, deep learning approaches, and the transformer-based approaches. In addition, this chapter illustrated the usage of the latest transformer-based approaches in modern Arabic sentiment analysis researches on social media, and dialectal Arabic. It also discussed about the metrics of performance in sentiment classification as they are important to evaluate the performance of models and compare between different approaches.

Moreover, the studies reveal that there are many recent approaches achieved better results by using deep learning and transformer-based models. However, most of these studies do not exactly fit with the current task, so a focused and practical framework for Arabic tweet sentiment classification is still requiring. Based on the review, the domain-adaptive transformer feature extraction and machine learning classification applied on the ASAD dataset is used in the proposed project. Traditional machine learning models are used as baselines, MARBERTv2 is used as the main transformer model. Overall, this chapter provides the backgrounds for the methodology in the next chapter.

Chapter 3 Proposed Methodology

3.1 Introduction

This chapter presents the proposed approach for binary sentiment classification of Arabic tweets. It describes the complete experimental workflow, beginning with dataset preparation and preprocessing and continuing through feature representation, model training, and performance evaluation. The approach was designed to compare different modeling strategies under a consistent experimental setting.

3.2 Proposed Framework

The proposed framework outlines the main stages followed in this study for Arabic sentiment classification. As illustrated in Figure 10, the framework begins with the ASAD dataset and proceeds through data preparation, text preprocessing, feature representation, classification, and performance evaluation. Neutral tweets are excluded to formulate the task as binary classification between positive and negative sentiments. Three modeling strategies are evaluated. The first uses character-level TF-IDF features with Logistic Regression and Support Vector Machine. The second combines contextual CLS representations extracted from MARBERTv2 with the same classifiers. The third evaluates neural adaptation approaches, including fine-tuned MARBERTv2 and two CNN-based configurations. All models use the same data partitions and are evaluated using accuracy, precision, recall, and F1-score. The subsequent sections explain each stage of the framework in detail.

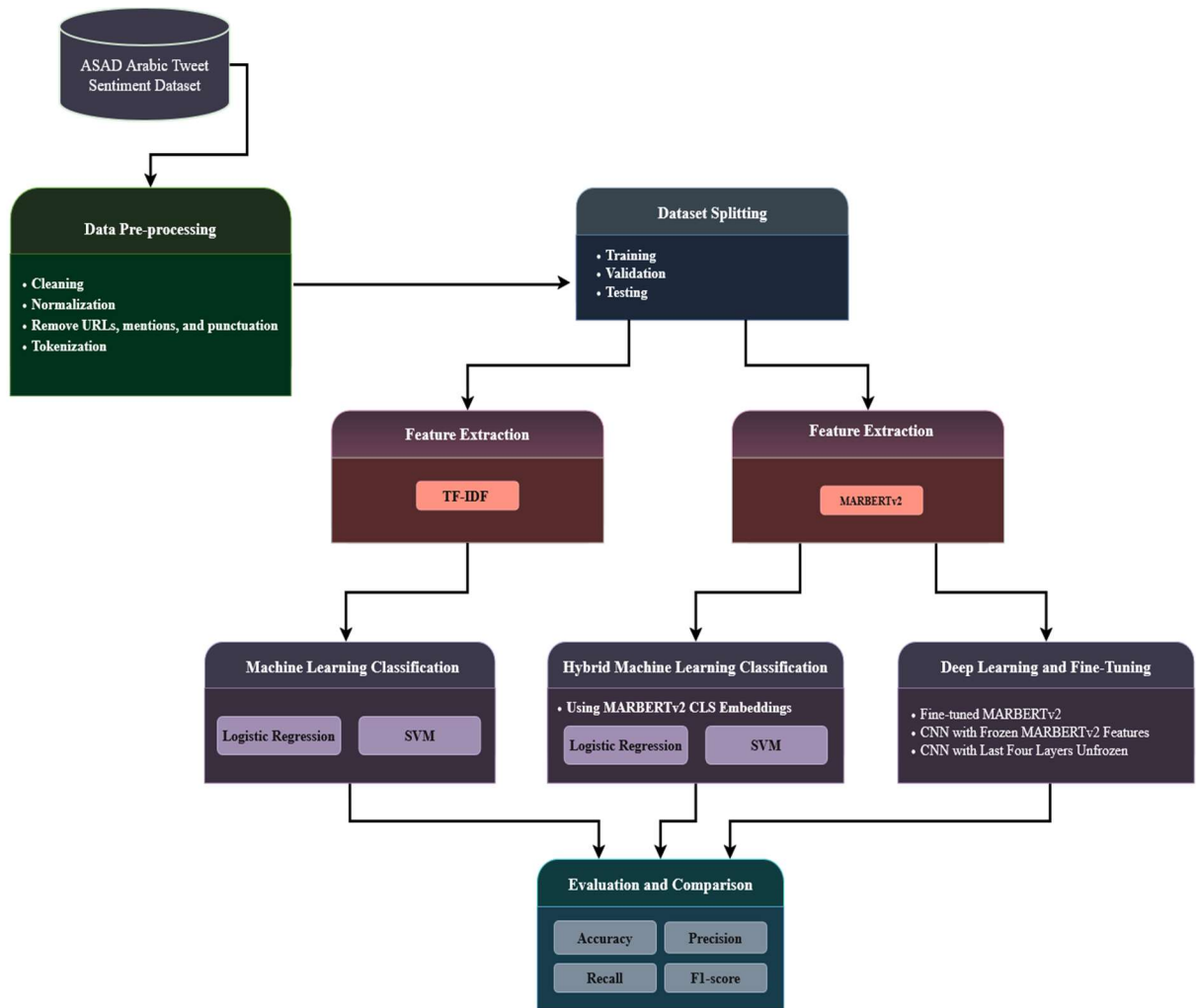


Figure 10. Proposed Framework.

3.3 Dataset Description

This study uses the Arabic Sentiment Analysis Dataset (ASAD), a large benchmark dataset developed for sentiment classification of Arabic tweets [47]. The original dataset contains approximately 100,000 tweets labeled as positive, negative, or neutral. The tweets were collected between May 2012 and April 2020 and include Modern Standard Arabic and several Arabic dialects, such as Khaleeji, Hijazi, and Egyptian Arabic. For this study, the train_all_ext.csv and test1_labeled.csv files were combined. Missing values and tweets with duplicate Tweet IDs were removed, and neutral tweets were excluded to formulate the task as binary sentiment classification between positive and negative sentiments. After these steps, the final dataset contained 26,785 tweets, including 13,401 positive and 13,384 negative instances. Figure 11 illustrates distribution of Positive and Negative Tweets Across the Dataset Partitions.

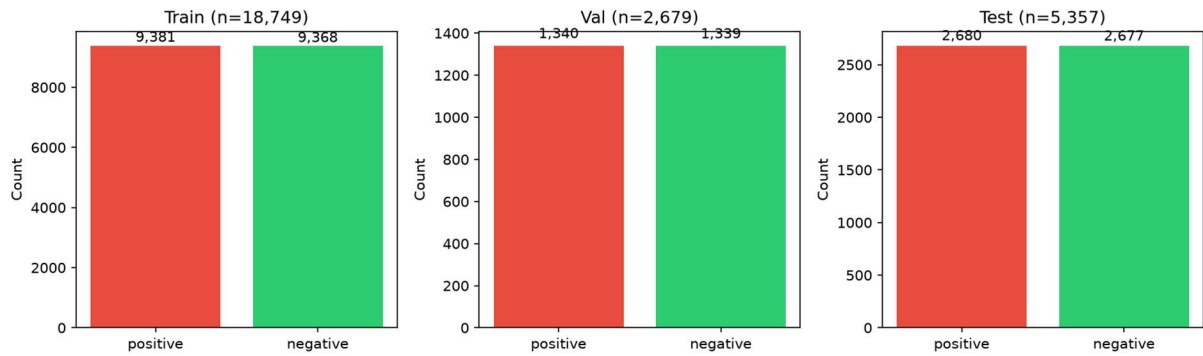


Figure 11. Distribution of the dataset.

3.4 Data Preprocessing

Data preprocessing was applied to improve the quality of the Arabic tweets before training the models. URLs, user mentions, hashtags, retweet markers, and unwanted symbols were removed. Sentiment-related emojis were converted into Arabic words, such as حب, سعيد, حزين, and غاضب. Some Arabizi symbols were also converted into their corresponding Arabic letters. Arabic letters were normalized, while diacritics and Tatweel were removed. Repeated letters and punctuation marks were reduced, numbers were converted into Arabic words, and extra spaces were removed. Stemming, lemmatization, and stop-word removal were not applied to preserve the meaning and dialectal information of the tweets. Table 3 presents examples of Arabic tweets before and after preprocessing.

Table 3. Examples of Arabic Tweets Before and After Preprocessing.

Before Preprocessing	After Preprocessing
موسم جدّة # ❤️ هرمنا لأجل هذا اليوم 😊😊 أخیییییییییرا @SJofficial https://t.co/XeRpXuBHY6	اخیرا حب حب هرمنا لأجل هذا اليوم حزين حب
دایم تبهرونا بالتمیز فی کل التفاصيل @Turki_alalshikh 🤝❤️ الوعد الرياض #	دایم تبهرونا بالتمیز فی کل التفاصيل حب تصفیق
شکراً وزارة الثقافة 🤝❤️❤️ أحلی شی فی أزمة كورونا کلها https://t.co/jTdYX0VOE1	احلی شی فی ازمه كورونا کلها حب حب حب شکرا وزارة الثقافة حب

To better understand the processed tweets, word clouds were generated from the positive and negative training tweets. Figure 12 shows the most frequent words in both sentiment classes after preprocessing. The positive word cloud includes words such as شكرًا, حب, جميل, while the negative word cloud contains common dialectal words such as مش, ليش, يعني. Common Arabic and dialectal words remained because stemming and stop-word removal were not applied.



Feature representation is the process of converting processed text into numerical features that classification models can understand and use. In this study, three feature representations were used according to the requirements of the classification models. Character-level Term Frequency–Inverse Document Frequency (TF-IDF) features were used for the traditional machine learning models, MARBERTv2 CLS features were used for the machine learning models, and MARBERTv2 token representations were used for the neural models.

TF-IDF is a statistical method that converts text into numerical weights based on the importance of textual patterns in the dataset. A pattern receives a higher weight when it appears frequently in a particular tweet but is less common across the other tweets. In this study, character n-grams within word boundaries were used to capture short textual patterns, spelling variations, and dialectal forms in Arabic tweets. The TF-IDF vectorizer was fitted using the training tweets and then applied to the validation and test tweets. The extracted sparse numerical features were used as input for Logistic Regression and Support Vector Machine. The TF-IDF configuration used in this study is presented in Table 4.

26

Setting	Value
Analyzer	Character n-grams within word boundaries
N-gram range	2–5
Maximum features	150,000
Minimum document frequency	2
Sublinear term frequency	Enabled

3.5.2 MARBERTv2 Model

For the machine learning , the CLS token is a special token added at the beginning of each input sequence. Its final hidden representation provides a single numerical vector that summarizes the contextual meaning of the entire tweet. MARBERTv2 was used to extract this representation because it was developed for Arabic dialects and social media text [26]. Each tweet was tokenized and padded or truncated to a maximum sequence length of 128 tokens. The final hidden representation of the CLS token was then extracted as a 768-dimensional feature vector. These contextual features were used as input for Logistic Regression and Support Vector Machine, forming the machine learning approach.

3.5.3 MARBERTv2 Token Representations for Neural Models

Token representation assigns a contextual numerical vector to every token in a tweet instead of representing the entire tweet with one vector. This allows neural models to identify important local patterns and relationships among the tokens. Each tweet was represented as a sequence of 128 token vectors, with 768 features for each token. For the TextCNN model using frozen MARBERTv2 features, the token representations were extracted before training and remained unchanged. For the CNN model combined with the last four MARBERTv2 layers, the token representations were updated during training. Table 5 summarizes the MARBERTv2 representation configuration used in this study.

Table 5. MARBERTv2 parameters

Setting	Value
Pre-trained model	UBC-NLP/MARBERTv2
Maximum sequence length	128 tokens
Padding	Maximum length
Truncation	Enabled
CLS feature vector size	768

All-token representation size	128 × 768
--------------------------------------	------------------

3.6 Machine Learning Models

This section presents the machine learning models evaluated using the feature representations described in Section 3.5. Logistic Regression and Linear Support Vector Machine (Linear SVM) were trained separately using character-level TF-IDF features and 768-dimensional MARBERTv2 CLS features. Accordingly, four model configurations were evaluated. Table 6 presents the hyperparameters used for these models.

Table 6. Hyperparameters of the Machine Learning Models

Feature Representation	Model	C	Maximum Iterations	Calibration
Character-level TF-IDF	Logistic Regression	5.0	1,000	—
Character-level TF-IDF	Linear SVM	0.5	2,000	Three-fold
MARBERTv2 CLS Features	Logistic Regression	0.5	1,000	—
MARBERTv2 CLS Features	Linear SVM	0.1	2,000	Three-fold

The regularization parameter (C) was selected using GridSearchCV based on the validation F1-score. A PredefinedSplit was used to preserve the predefined training and validation partitions during model selection. After selecting the optimal value, each classifier was retrained using the combined training and validation sets. CalibratedClassifierCV with three-fold cross-validation was applied to the Linear SVM models to generate class probabilities. A random seed of 42 was used to support the reproducibility of the experiments.

3.7 Deep Learning Models

Deep learning models use neural network layers to learn complex patterns from data. This study used a Convolutional Neural Network (CNN), which applies filters to identify important local patterns in token sequences. Two CNN configurations were evaluated using the MARBERTv2 token representations described in Section 3.5.3.

3.7.1 CNN with Frozen MARBERTv2 Features

Frozen features are representations extracted from a pre-trained model without updating its parameters during classifier training. In this model, MARBERTv2 was used as a fixed feature extractor, while only the CNN classifier was trained. Each tweet was represented by a 128×768 matrix. Four one-dimensional convolutional layers with kernel sizes of 2, 3, 4, and 5 were used to capture patterns of different lengths. Each layer contained 256 filters followed by the Rectified Linear Unit (ReLU), which retains positive values. Global max pooling selected the most important feature from each filter. The outputs were then combined and passed through dropout and a two-unit classification layer. The architecture is presented in Table 7.

Table 7. Architecture of CNN with Frozen MARBERTv2 Features.

Layer	Configuration	Activation	Description
Input	128×768	—	Frozen MARBERTv2 token features
Convolution	256 filters with kernel sizes 2, 3, 4, and 5	ReLU	Extracts local textual patterns
Global Max Pooling	One value per filter	—	Retains the most important features
Concatenation	1,024 features	—	Combines the convolution outputs
Dropout	0.50	—	Reduces overfitting
Output	2 units	—	Predicts positive or negative sentiment

The model was trained using the Adam optimizer with a learning rate of 1×10^{-3} , a batch size of 64, and cross-entropy loss. CosineAnnealingLR gradually adjusted the learning rate. Training continued for a maximum of 100 epochs, with early stopping after six epochs without improvement in the validation F1-score.

3.7.2 CNN with the Last Four MARBERTv2 Layers Unfrozen

Partial fine-tuning updates selected layers of a pre-trained model while keeping the remaining layers frozen. In this model, the last four MARBERTv2 encoder layers were updated during training, allowing the token representations to adapt to the sentiment classification task [26]. The CNN used the same kernel sizes and number of filters as the frozen model. Batch normalization was applied to stabilize the convolution outputs, while dropout was used to reduce overfitting. Table 8 presents the architecture.

Table 8. Architecture of the CNN with the Last Four MARBERTv2 Layers Unfrozen.

Layer	Configuration	Activation	Description
MARBERTv2	Last four encoder layers unfrozen	—	Produces task-adaptive token features
Convolution	256 filters with kernel sizes 2, 3, 4, and 5	ReLU	Extracts local textual patterns
Batch Normalization	Applied after convolution	—	Stabilizes training
Global Max Pooling	One value per filter	—	Retains the most important features
Concatenation	1,024 features	—	Combines the convolution outputs
Dropout	0.60	—	Reduces overfitting
Output	2 units	—	Predicts positive or negative sentiment

Mixout with a probability of 0.90 was used to limit large changes in the pre-trained weights, while label smoothing of 0.10 reduced overconfident predictions. The model was trained using AdamW with a learning rate of 7×10^{-6} and a batch size of 16. A cosine scheduler with a 10% warmup period and gradient clipping of 1.0 were applied. Early stopping was used after 15 epochs without improvement in the validation F1-score.

3.8 Transformer-Based Models

Transformer-based models are pre-trained language models that learn contextual relationships between tokens in a text. Unlike traditional models, they can represent the meaning of each token according to its surrounding context. This study used MARBERTv2 because it was pre-trained on Arabic social media text and is suitable for processing Arabic dialects [26].

3.8.1 MARBERTv2 Fine-Tuning

MARBERTv2 was fine-tuned end-to-end for binary sentiment classification using positive and negative labels. The input tweets were tokenized, truncated, and padded to a maximum sequence length of 128 tokens. Table 9 presents the fine-tuning configuration used in the experiment.

Table 9. MARBERTv2 Fine-Tuning Configuration

Parameter	Value
Pre-trained model	UBC-NLP/MARBERTv2

Number of labels	2
Maximum sequence length	128 tokens
Learning rate	1.2×10^{-5}
Training batch size	8
Evaluation batch size	16
Maximum number of epochs	10
Weight decay	0.01
Warmup ratio	0.1
Learning-rate scheduler	Cosine
Evaluation and saving strategy	At the end of each epoch
Best-model selection metric	Validation F1-score
Early-stopping patience	3 epochs
Random seed	42

The model was evaluated on the validation set after each epoch. The checkpoint with the highest validation F1-score was retained, and training was stopped when the F1-score did not improve for three consecutive epochs. The selected checkpoint was then evaluated on the held-out test set.

3.9 Chapter Summary

This chapter presents the methodology used for binary Arabic sentiment classification. It began by describing the proposed framework and the ASAD dataset. Only positive and negative tweets were selected, and the data was divided into training, validation, and test sets using stratified sampling. The preprocessing steps were then explained, including text normalization, removal of noise, and treatment of Arabic letters and social media elements. Three feature representations were used: character-level TF-IDF features, MARBERTv2 CLS features, and contextual token representations. Logistic Regression and SVM were evaluated using TF-IDF and MARBERTv2 CLS features. Two CNN configurations were also applied: one using frozen MARBERTv2 token features and another with the last four MARBERTv2 layers unfrozen. In addition, MARBERTv2 was directly fine-tuned for sentiment classification. The validation set was used for model selection and early stopping, while the test set was reserved for final evaluation. Model performance was assessed using the accuracy, precision, recall, and F1-score metrics described in Section 2.2.5. The next chapter presents and discusses the experimental results.

Chapter 4 Experimental Results and Discussion

4.1 Introduction

This chapter presents the results obtained after applying the proposed binary Arabic sentiment classification framework to the ASAD dataset. Three model families are examined: machine learning classifiers, deep learning architectures, and transformer-based models. All models were developed using the same held-out test set to ensure a fair comparison. Their performance was assessed using accuracy, precision, recall, and F1-score. The results are organized according to the model categories, followed by a comparison and discussion of the main findings, the study limitations, and a summary of the chapter.

4.2 Experimental Setup

The experiments were conducted using the binary version of the ASAD dataset, which contained 26,785 tweets: 13,401 positive and 13,384 negative. Using stratified sampling with a random seed of 42, the dataset was divided into 18,749 training samples (70%), 2,679 validation samples (10%), and 5,357 testing samples (20%). The experiments included Logistic Regression and Linear SVM using character-level TF-IDF and MARBERTv2 CLS features, a TextCNN model using frozen MARBERTv2 representations, a CNN model with the last four MARBERTv2 layers unfrozen, and a fully fine-tuned MARBERTv2 model. The experiments were implemented in a Jupyter Notebook using Python and the NumPy, Pandas, Scikit-learn, PyTorch, and Hugging Face Transformers libraries. PyTorch 2.13.0 was used with Metal Performance Shaders (MPS) as the computational device. The best hyperparameters and model checkpoints were selected based on the validation F1-score. All models were evaluated using the same held-out test set to ensure a fair comparison, and their final performance was measured using accuracy, precision, recall, and F1-score.

4.3 Results of Machine Learning Models

The machine learning classifiers were evaluated using two feature representations. The traditional models used character-level TF-IDF features, while the models used the contextual CLS features extracted from MARBERTv2. Logistic Regression and SVM were tested with both representations. Table 10 summarizes their performance on the test set.

Table 10. Performance of ML Models

Feature Representation	Model	Accuracy	Precision	Recall	F1-score
Character-level TF-IDF	Logistic Regression	0.9061	0.9163	0.8940	0.9050
Character-level TF-IDF	SVM	0.9076	0.9146	0.8993	0.9069
MARBERTv2 CLS Features	Logistic Regression	0.9050	0.9107	0.8981	0.9044
MARBERTv2 CLS Features	SVM	0.9055	0.9127	0.8970	0.9048

4.4 Results of Deep Learning Models

Two CNN models were evaluated using the contextual token representations generated by MARBERTv2. The first model used frozen MARBERTv2 features, while the second model allowed the last four MARBERTv2 layers to be updated during training. Table 11 presents their performance on the test set. Figure 13 presents the training and validation learning curves of the CNN model with the last four MARBERTv2 layers unfrozen.

Table 11. Performance of Deep Learning Models on the ASAD Test Set.

Model	Accuracy	Precision	Recall	F1-score	Selected Epoch
CNN with Frozen MARBERTv2 Features	0.9406	0.9394	0.9422	0.9408	12
CNN with the Last Four MARBERTv2 Layers Unfrozen	0.9520	0.9382	0.9679	0.9528	31

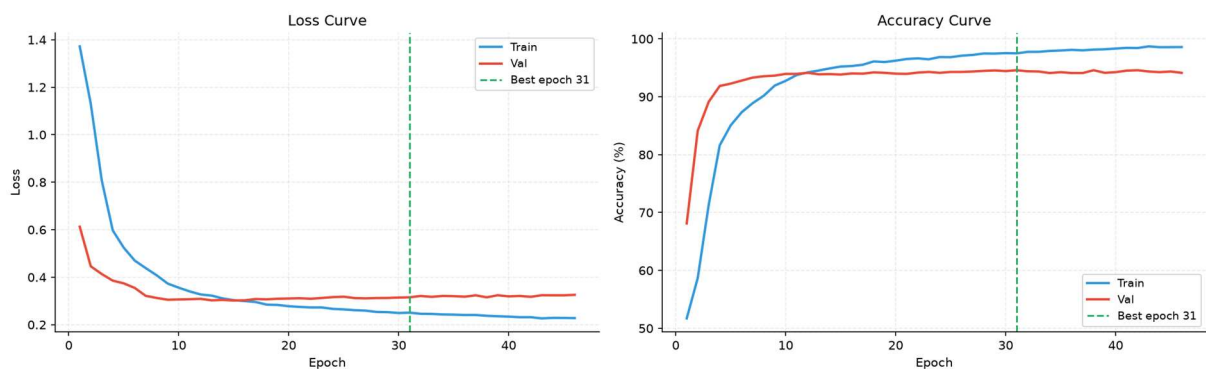


Figure 13. Training and Validation Learning Curves of the Partially Fine-Tuned CNN Model.

The learning curves show that the training performance continued to improve across epochs, while the validation performance reached its best level around epoch 31. Therefore, the checkpoint obtained at epoch 31 was selected for the final test evaluation. Figure 14 illustrates the difference between training and validation performance during the training process.

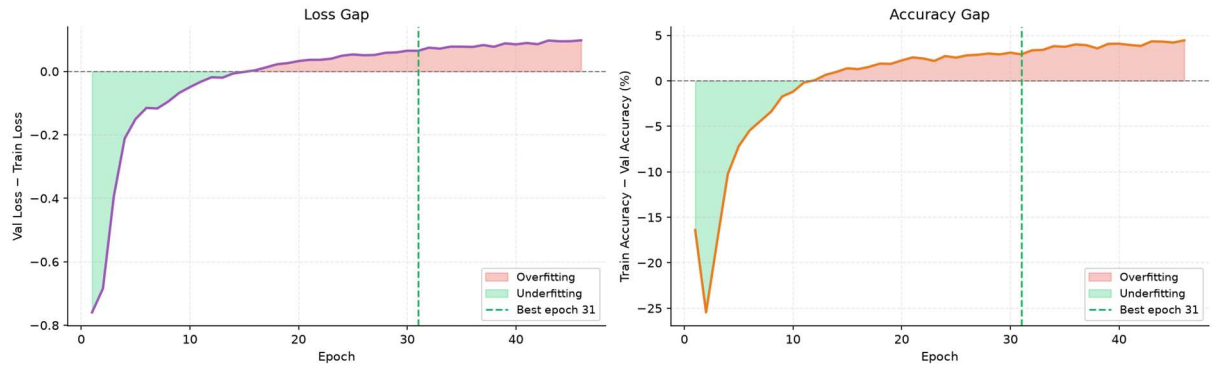


Figure 14. Overfitting Analysis of the Partially Fine-Tuned CNN Model.

After the selected epoch, the training performance continued to improve while the validation performance showed limited improvement and the validation loss began to increase slightly. This indicates the beginning of limited overfitting. Early stopping and selecting the best checkpoint helped prevent the use of a later and more overfitted model. Figure 15 presents the confusion matrix of the CNN model with the last four MARBERTv2 layers unfrozen. The model correctly classified 2,506 negative tweets and 2,594 positive tweets. It misclassified 171 negative tweets as positive and 86 positive tweets as negative. The lower number of false negatives is consistent with the model's high recall of 0.9679.

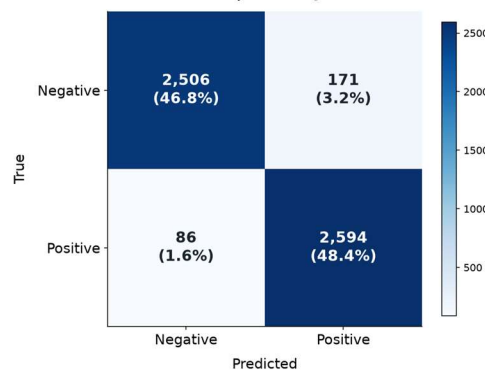


Figure 15. Confusion Matrix of the CNN with the Last Four MARBERTv2 Layers Unfrozen.

4.5 Results of Transformer-Based Models

MARBERTv2 was fine-tuned directly on the binary ASAD training set to classify tweets as positive or negative. The checkpoint with the highest validation F1-score was selected and evaluated on the test set. Table 12 presents the performance of the fine-tuned model.

Table 12. Performance of the Transformer-Based Model on the ASAD Test Set.

Model	Accuracy	Precision	Recall	F1-score
Fine-tuned MARBERTv2	0.9526	0.9473	0.9586	0.9529

4.6 Discussion of Findings

The machine learning models produced closely comparable results. Logistic Regression with character-level TF-IDF achieved the highest precision among these models, whereas SVM with character-level TF-IDF achieved the highest accuracy and F1-score of 0.9076 and 0.9069, respectively. This indicates that SVM provided a slightly better balance between precision and recall in the high-dimensional and sparse TF-IDF feature space. The two models based on fixed MARBERTv2 CLS features achieved similar but slightly lower results. Although the CLS representation contained contextual information, representing the entire tweet with one fixed vector may not have preserved all sentiment-related patterns required by the linear classifiers. In contrast, character-level TF-IDF effectively represented spelling variations, word fragments, and morphological patterns commonly found in Arabic tweets. Combining token-level MARBERTv2 representations with CNN substantially improved the results. The CNN with frozen MARBERTv2 features achieved an accuracy of 0.9406 and an F1-score of 0.9408, outperforming all traditional machine learning models. This improvement may be attributed to the CNN's ability to process multiple contextual token representations and capture local sentiment patterns. Unfreezing the final four MARBERTv2 layers further increased the accuracy to 0.9520 and the F1-score to 0.9528. Its recall increased to 0.9679, although its precision decreased slightly compared with the frozen model. This shows that partial fine-tuning made the model more sensitive to positive sentiment, allowing it to identify more positive tweets while producing slightly more false-positive predictions. The learning curves and confusion matrix support these findings. Validation performance reached its best level around epoch 31, after which the training performance continued to improve while the validation loss began to increase slightly. This indicated the beginning of limited overfitting; therefore, the best checkpoint and early stopping were used to preserve generalization. The partially fine-tuned CNN correctly classified 2,506 negative and 2,594 positive tweets. It produced 171 false positives and only 86 false negatives, explaining why it achieved the highest recall among

all evaluated models. Fine-tuned MARBERTv2 achieved the best overall accuracy of 0.9526 and F1-score of 0.9529. Compared with the partially fine-tuned CNN, it achieved higher precision but slightly lower recall, resulting in a more balanced performance. However, the difference between the two models was minimal, showing that adapting only the final four MARBERTv2 layers with CNN was nearly as effective as fine-tuning the entire transformer. Overall, fine-tuned MARBERTv2 improved accuracy by 4.50 percentage points and F1-score by 4.60 percentage points compared with TF-IDF with SVM. These findings demonstrate that MARBERTv2 provided its greatest benefit when its contextual representations were partially or fully adapted to the sentiment classification task. Compared with Mansour et al. [41], the transformer ensemble reported an accuracy of 0.9560 and an F1-score of 0.9500. Fine-tuned MARBERTv2 achieved slightly lower accuracy but a higher F1-score of 0.9529 using a single transformer model. The partially fine-tuned CNN also achieved a higher F1-score of 0.9528. Nevertheless, these small differences should be interpreted cautiously because preprocessing procedures, data splitting, and training configurations may differ between the two studies. Figure 16 summarizes the performance of all evaluated models and confirms that partial and full adaptation of MARBERTv2 produced the strongest results.

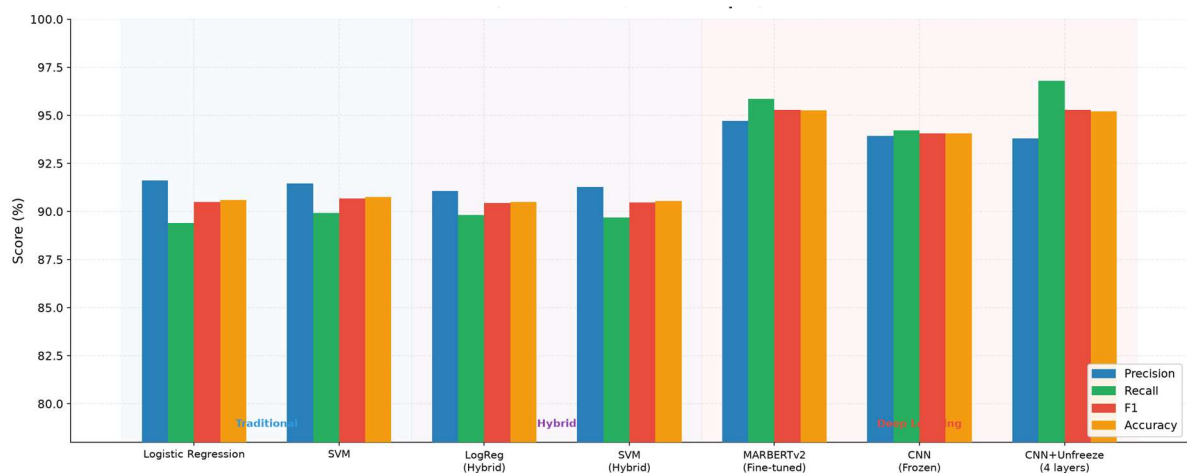


Figure 16. Performance Comparison of All Evaluated Models.

4.7 Limitations

Despite the promising results, this study has several limitations. First, the experiments were conducted using only the ASAD dataset, which may limit the generalizability of the findings to other datasets. Second, ASAD consists mainly of Arabic tweets; therefore, the results may not fully generalize to other domains, text types, or Arabic dialects. Third, the neutral class was excluded, and the experiments focused on a balanced binary classification task involving only positive and negative sentiments. Consequently, the findings may not fully represent real-world sentiment distributions or more complex three-class classification settings that include positive, negative, and neutral sentiments.

4.8 Chapter Summary

This chapter presented and discussed the experimental results of machine learning, deep learning, and transformer-based models. The findings showed that adapting MARBERTv2 to the sentiment classification task improved the overall performance compared with traditional models and frozen feature extraction. The results were also compared with the reference study, and the main limitations of the experiments were discussed. The next chapter presents the conclusion and recommendations for future work.

Chapter 5 Conclusion and Future Work

This thesis investigated an enhanced framework for binary Arabic sentiment analysis by examining the effectiveness of domain-adaptive transformer representations and their integration with machine learning and deep learning classification approaches. The proposed framework was evaluated using the Arabic Sentiment Analysis Dataset (ASAD), with neutral tweets excluding formulating the task as binary classification between positive and negative sentiments. The dataset was divided using stratified sampling into training, validation, and testing sets to provide a consistent basis for model comparison. The experimental framework compared traditional machine learning, hybrid machine learning, CNN-based models, and a directly fine-tuned transformer model. Character-level TF-IDF features were used with Logistic Regression and SVM as traditional baselines, while MARBERTv2 CLS representations were combined with the same classifiers to investigate the effect of contextual transformer features. In addition, two CNN configurations were evaluated using MARBERTv2 token representations: a CNN with frozen transformer features and a CNN in which the last four MARBERTv2 layers were unfrozen. Finally, MARBERTv2 was directly fine-tuned for binary sentiment classification. All models were evaluated using the same data partitions and the same evaluation metrics, including accuracy, precision, recall, and F1-score.

The results demonstrate that transformer adaptation provides a clear advantage over the traditional machine learning baselines. The directly fine-tuned MARBERTv2 model achieved the best overall performance. It obtained an accuracy of 95.26%, precision of 94.73%, recall of 95.86%, and F1-score of 95.29%. The results therefore support the use of domain-adaptive transformer models as an effective direction for improving Arabic sentiment classification. Several directions can be considered for future research. First, the proposed models should be evaluated on additional Arabic sentiment datasets to determine

whether the observed improvements are consistent across different datasets and linguistic contexts. Second, future studies should investigate cross-domain Arabic sentiment classification. Since Arabic sentiment expressions vary across tweets, reviews, online discussions, and other forms of user-generated content, evaluating the models across different domains would provide a more realistic assessment of their robustness. Third, additional Arabic dialects and multilingual or code-switching settings could be investigated. This would help determine whether the proposed framework can effectively handle the linguistic diversity and variation that characterize Arabic social media communication.

References

- [1] Shi, Z., AlShenaifi, N., Al-Refaie, H., Abdul-Mageed, M., & Ungar, L. (2025). *A comprehensive survey of contemporary Arabic sentiment analysis: Methods, challenges, and future directions*. In *Findings of the Association for Computational Linguistics: NAACL 2025*. Association for Computational Linguistics.

- [2] El Mekki, A., El Mahdaouy, A., Berrada, I., & Khoumsi, A. (2021). *Domain adaptation for Arabic cross-domain and cross-dialect sentiment analysis from contextualized word embedding*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.
- [3] Fang, Y., Xu, C., Guan, S., Yan, N., & Mei, Y. (2024). *Advancing Arabic sentiment analysis: ArSen benchmark and the improved fuzzy deep hybrid network*. In *Proceedings of the 28th Conference on Computational Natural Language Learning*. Association for Computational Linguistics.
- [4] Alfugaha, R., & AL-Smadi, M. (2025). *A hybrid transformer-based model for sentiment analysis of Arabic dialect hotel reviews*. In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*. INCOMA Ltd.
- [5] Liaqat, M. I., et al. (2022). *Sentiment analysis techniques, challenges, and opportunities*.
- [6] Matrane, Y., et al. (2023). *A systematic literature review of Arabic dialect sentiment analysis*.
- [7] Taher, H. A. (2024). *Dual-Language Sentiment Analysis: A Comprehensive Evaluating SVM, Logistic Regression, XGBoost, and Decision Tree Using TF-IDF on Arabic and English Dataset*.
- [8] Aljabri, M., Chrouf, S. M. B., Alzahrani, N. A., Alghamdi, L., Alfehaid, R., Alqarawi, R., Alhuthayfi, J., & Alduhailan, N. (2021). *Sentiment Analysis of Arabic Tweets Regarding Distance Learning in Saudi Arabia during the COVID-19 Pandemic*.
- [9] Jha, U. M., & Guhan, B. (2024). *A comparison of the classification models of machine learning for predicting the risks of diabetes: An analysis of National Family Health Survey data of Delhi*.
- [10] Alyami, S. N., & Olatunji, S. O. (2020). *Application of Support Vector Machine for Arabic Sentiment Classification Using Twitter-Based Dataset*.
- [11] Cardoso-Fernandes, J., Teodoro, A. C., Lima, A., & Roda-Robles, E. (2020). *Semi-Automatization of support vector machines to map lithium (Li) bearing pegmatites*.
- [12] Alghamdi, H. M. (2024). *Unveiling Sentiments: A Comprehensive Analysis of Arabic Hajj-Related Tweets from 2017–2022 Utilizing Advanced AI Models*.
- [13] Chawla, A. (2024, August 14). *Formulating and implementing XGBoost from scratch*.

- [14] Al-Gaphari, G., et al. (2025). *Enhancing Arabic sentiment analysis during the Qatar World Cup: A study of generative pre-trained transformers and machine learning techniques*.
- [15] Mhamed, M., Sutcliffe, R., & Feng, J. (2025). *Benchmark Arabic news posts and analyzes Arabic sentiment through RMuBERT and SSL with AMCFFL technique*. Egyptian Informatics Journal, 29, 100601.
- [16] Elhassan, N., Varone, G., Ahmed, R., et al. (2023). *Arabic Sentiment Analysis Based on Word Embeddings and Deep Learning*.
- [17] Khamket, T., & Polpinij, J. (2022). *Automatically correcting noisy labels for improving quality of training set in domain-specific sentiment classification*.
- [18] Basiri, M. E., Nemati, S., Abdar, M., Asadi, S., & Acharya, U. R. (2021). *ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis*.
- [19] Xu, Z. (2023). *Sentiment analysis of hotel comments based on LSTM and GRU*. In *Proceedings of the 2023 International Conference on Machine Learning and Automation*.
- [20] Alhumoud, S. O. (2024). *CNN-BiGRU-Focus: A Hybrid Deep Learning Classifier for Sentiment and Hate Speech Analysis of Ashura-Arabic Content for Policy Makers*.
- [21] Harrou, F., Dairi, A., Dorbane, A., & Sun, Y. (2023). *Energy consumption prediction in water treatment plants using deep learning with data augmentation*.
- [22] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). *Deep learning-based text classification: A comprehensive review*.
- [23] Mutawa, A. M., et al. (2025). *A Comparative Evaluation of Transformers and Deep Learning Models for Arabic Text Classification*. Applied Sciences, 15(9), 4941.
- [24] Antoun, W., Baly, F., & Hajj, H. (2020). *AraBERT: Transformer-based model for Arabic language understanding*.
- [25] Abd Elaziz, M. E., Dahou, A., Orabi, D. A., Awadallah, M. A., Lu, S., Al-Qaness, M. A. A., & Ewees, A. A. (2023). *A hybrid multitask learning framework with a Fire Hawk optimizer for Arabic fake news detection*.
- [26] Abdul-Mageed, M., Elmadany, A., Nagoudi, E. M. B., Pabbi, D., Verma, K., & Lin, R. (2021). *ARBERT & MARBERT: Deep bidirectional transformers for Arabic*.
- [27] Gabr, M. A.-E. I. I., Badr, A. Z., & Mahdi, H. M. K. (2023). *Exploring strategies for measuring semantic similarity in short Arabic texts*. Indian Journal of Computer Science and Engineering, 14(3), 480–489.

- [28] Inoue, G., Alhafni, B., Baimukan, N., et al. (2021). *The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models*.
- [29] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*.
- [30] Jurafsky, D., & Martin, J. H. (2025). *Speech and language processing* (3rd ed., draft). Stanford University & University of Colorado Boulder.
- [31] Terven, J., Cordova-Esparza, D.-M., Romero-González, J.-A., Ramírez-Pedraza, A., & Chávez-Urbiola, E. A. (2025). *A comprehensive survey of loss functions and metrics in deep learning*. *Artificial Intelligence Review*, 58(7).
- [32] Abdul-Mageed, M., Zhang, C., Elmadany, A., Bouamor, H., & Habash, N. (2022). *NADI 2022: The third nuanced Arabic dialect identification shared task*.
- [33] Abdel-Salam, R. (2022). *Dialect & sentiment identification in nuanced Arabic tweets using an ensemble of prompt-based, fine-tuned, and multitask BERT-based models*.
- [34] Bayrak, G., & Issifu, A. M. (2022). *Domain-adapted BERT-based models for nuanced Arabic dialect identification and tweet sentiment analysis*.
- [35] Qaddoumi, A. (2022). *Arabic sentiment ensemble NADI shared task 2*.
- [36] Belbachir, F. (2023). *Foul at SemEval-2023 Task 12: MARBERT language model and lexical filtering for sentiment analysis of tweets in Algerian Arabic*.
- [37] Alosaimi, W., Saleh, H., Hamzah, A. A., El-Rashidy, N., Alharb, A., Elaraby, A., & Mostafa, S. (2024). *ArabBert-LSTM: Improving Arabic sentiment analysis based on transformer model and long short-term memory*.
- [38] Baniata, L. H., & Kang, S. (2024). *Switch-transformer sentiment analysis model for Arabic dialects that utilizes a mixture of experts mechanism*.
- [39] Almaqtari, H., Zeng, F., & Mohammed, A. (2024). *Enhancing Arabic sentiment analysis of consumer reviews: Machine learning and deep learning methods based on NLP*.
- [40] Alawneh, H., Hasasneh, A., & Maree, M. (2024). *On the utilization of emoji encoding and data preprocessing with a combined CNN-LSTM framework for Arabic sentiment analysis*.
- [41] Mansour, O., Aboelela, E. M., Talaat, R., & Bustami, M. (2025). *Transformer-based ensemble model for dialectal Arabic sentiment classification*.

- [42] Aljomah, F., Aldhafeeri, L., Alfadel, M., Alshahrani, S., Abbas, Q., & Alhumoud, S. (2025). *Enhancing Arabic sentiment analysis with pre-trained CAMELBERT: A case study on noisy texts*.
- [43] Jaber, A., Bahati, I., & Martínez, P. (2025). *Leveraging pre-trained embeddings in an ensemble machine learning approach for Arabic sentiment analysis*.
- [44] Alharbi, M. I., Chafik, S., Ezzini, S., Mitkov, R., Ranasinghe, T., & Hettiarachchi, H. (2025). *AHaSIS: Shared task on sentiment analysis for Arabic dialects*.
- [45] Alghamdi, S., Rayson, P., & Alotibi, R. (2026). *Improving on state-of-the-art models for sentiment analysis on Saudi-English code-switching text*.
- [46] Alsahafi, S., Alshutayri, A., & Alahdal, S. (2026). *Sentiment analysis of tweets during the Riyadh Season using deep learning*.
- [47] Alamro, H., Alshehri, M., Alharbi, B., Khayyat, Z., Kalkatawi, M., Jaber, I. I., & Zhang, X. (2021). *Overview of the Arabic Sentiment Analysis 2021 Competition at KAUST*. arXiv preprint arXiv:2109.14456.