Arab American University

Faculty of Graduate Studies

A Hybrid CNN-LSTM Framework for Enhanced Arabic Sentiment Analysis: Investigating Emoji Encoding and Preprocessing Strategies

By

Hussam Fawzi Abed Alawneh

Supervisor

Dr. Ahmad Hasasneh

Co- Supervisor

Dr. Mohammed Maree

This thesis was submitted in partial fulfillment of the requirements for the Master's degree in Data Science and Business Analytics.

9/2025

I

Thesis Approval

A Hybrid CNN-LSTM Framework for Enhanced Arabic Sentiment Analysis: Investigating Emoji Encoding and Preprocessing Strategies

By

Hussam Fawzi Abed Alawneh

This thesis was defended successfully on 28.9.2025 and approved by:

Committee members	Signature
1. Dr. Ahmad Hasasneh: Supervisor	
2. Dr. Mohammed Maree: Co- Supervisor	
3. Dr. Ahmad Ewais: Internal Examiner	
4. Dr. Anas Tomah: External Examiner	

Declaration

I, Hussam Fawzi Abed Alawneh, declare that this thesis/dissertation is entirely my own work and that all sources used are properly acknowledged.

Student Name: Hussam Fawzi Abed Alawneh

University ID: 202113031

Signature: Hussam Alawneh

Date: 2.2.2026

Dedication

I dedicate this thesis to my beloved family, whose constant support and encouragement have been the foundation of my success. To my parents, your unwavering belief in my abilities throughout my academic journey has inspired me to push through every challenge. This thesis is a reflection of your guidance, love, and dedication. To my mentors, thank you for your insights and encouragement that helped me reach this milestone.

Acknowledgments

I want to express my sincere gratitude to my supervisors, Dr. Ahmad Hasasneh and Dr. Mohammed Maree, for their invaluable guidance, expertise, and continuous support throughout the process of developing this thesis. Their insightful feedback and support have been invaluable in overcoming challenges and producing a successful thesis.

Abstract

Social media users often express emotions, ideas, and thoughts through text in posts and tweets, which can be used to determine the text's polarity as positive or negative - a process known as sentiment analysis. Sentiment analysis has become critical for various real-world domains, including politics, tourism, e-commerce, education, and health. However, although sentiment analysis approaches perform well with English text, they face notable drawbacks when dealing with Arabic text. The morphological complexity inherent in the Arabic language poses challenges for building robust models, making it difficult to understand public sentiment and subsequently make informed decisions.

In response to these challenges, effective data preprocessing and deep learning techniques are employed to overcome the complexity of the Arabic language and provide insightful sentiment predictions. This thesis evaluates a combined Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) framework with different data preprocessing techniques for Arabic Sentiment Analysis (ASA) using the Arabic Sentiment Twitter Corpus (ASTC) dataset. Three experiments with eight distinct preprocessing configurations were conducted to evaluate the effect of data preprocessing on Arabic sentiment analysis, namely the effect of encoding and translating emojis to their real and emotional meanings. Emoji meanings were collected from four websites specialized in defining the meaning of emojis in social media, which resulted in a new dataset of emoji meaning called the "Emoji Meaning" dataset. Furthermore, the CNN-LSTM parameters were optimized using the Keras Tuner during the 5-fold cross-validation process.

The proposed model with emoji translation into Arabic text, obtained the highest accuracy rate (91.85%) by keeping non-Arabic words, removing punctuations, using the Snowball stemmer, and using Keras embedding. This approach yields competitive results compared to other state-of-the-art approaches, proves that emoji encoding enriches text by accurately reflecting emotions, and investigates the effect of data preprocessing on model performance. This allows the hybrid model to achieve results comparable to other studies that use the same ASTC dataset, thereby improving sentiment analysis accuracy.

Table of Contents

Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Explanation
ASA	Arabic Sentiment Analysis
ASTC	Arabic Sentiment Twitter Corpus
SA	Sentiment Analysis
NLP	Natural Language Processing
ML	Machine Learning
MSA	Modern Standard Arabic
CA	Classical Arabic
AD	Arabic Dialect
NN	Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
BiLSTM	Bidirectional Long Short-Term Memory
ARD	Arabic Reviews Dataset
GSC	Gold Standard Corpus
ASTD	Arabic Sentiment Tweets Dataset
GRU	Gated Recurrent Unit
BiGRU	Bidirectional Gated Recurrent Unit
HARD	Hotel Arabic-Reviews Dataset
BRAD	Books Reviews in Arabic Dataset
DNN	Deep Neural Networks
AJGT	Arabic Jordanian General Tweets
LARB	Large-Scale Arabic Book Reviews
SVM	Support Vector Machine
NB	Naive Bayes
RF	Random Forest
DE	Differential Evolution
KNN	K-Nearest Neighbor
LR	Logistic Regression
TF-IDF	Term Frequency - Inverse Document Frequency
BOW	Bag-of-Words
CBOW	Continuous Bag-of-Words
SS2030	Arabic Sentiment Analysis Dataset
DT	Decision Tree
MRMR	Minimum Redundancy Maximum Relevance

MNB	Multinomial Naive Bayes
MLP	Multi-Layer Perceptron
SGD	Stochastic Gradient Descent
CART	Classification And Regression Trees
CNB	Complementary Naive Bayes
SMO-SVM	Sequential Minimal Optimization-based Support Vector Machines
CAE	Correlation-Attribute Evaluator
ESR	Emoji Sentiment Ranking
PCA	Principal Component Analysis
Emo-SL	Emoji Sentiment Lexicon
LSA	Latent Semantic Analysis
MATTBiE-LSTM	Multi-level attention-based Bidirectional LSTM with bi-sense emoji embedding
WATT-BiE-LSTM	Word-guide Attention-based LSTM with Bi-sense Emoji Embedding
LLM	Large Language Model
USE	Universal Sentence Encoder
ANN	Artificial Neural Network
EDA	Exploratory Data Analysis
ISRI	Information Science Research Institute
HPO	Hyperparameter Optimization
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
TP	True Positive
TN	True Negative
FP	False-Positive
FN	False-Negative
TPR	True Positive Rate
FPR	False Positive Rate

Chapter 1 Introduction

1.1. Introduction

The field of Sentiment Analysis (SA) has gained significant attention for its ability to discern the feelings and opinions expressed on the internet, largely due to the proliferation and growth of social media platforms, which provide a large amount of textual data available on the internet [1]. SA, often referred to as opinion mining, is a critical type of Natural Language Processing (NLP) that classifies and analyzes textual data based on text polarity to understand sentiment [2]. SA plays a critical role in analyzing opinions, thoughts, and emotions expressed in tweets written about e-commerce, healthcare systems, and social networks [3] using the Twitter platform, now renamed X as of July 2023 [4]. This makes SA important for creating applications in diverse fields such as business analytics, analyzing customer feedback, monitoring social media posts, and tracking political sentiments. Thus, machine learning techniques that interpret human emotions and feelings in textual data provide a powerful tool for decision-makers in various aspects of life. Therefore, extending the application of SA to the Arabic language offers significant opportunities for Arabic communities in various domains.

Although Arabic is one of the most widely used languages in the world, research in Arabic sentiment analysis has progressed more slowly compared to research in other languages, such as English [5]. SA has been extensively explored in English language research, with comparatively less attention paid to the Arabic language [6]. Arabic sentiment analysis is a specialized field within NLP that aims to extract, understand, and classify sentiments, opinions, and emotions by analyzing texts written in the Arabic language as positive, negative, or neutral. Given that Arabic is the sixth most spoken language among all languages [7], with more than 500 million Arabic speakers globally [8], and about 185 million Arabic Web users [9], it is surprising that research in ASA is growing slowly. Thus, ASA is considered an active research field, especially utilizing Machine Learning (ML) techniques [10]. The increasing volume of Arabic content generated online, coupled with the ML capabilities, has created an urgent need to develop scalable and effective ASA techniques. However, the Arabic language faces several challenges that make ASA particularly complex.

1- **Diglossia**: The Arabic language has three forms, as shown in Figure 1.1.1. Firstly, Modern Standard Arabic (MSA) is the formal language used in official writing, education, and literature. Secondly, Classical Arabic (CA) is used in religious and literary contexts. Third is Arabic Dialect (AD), which varies from country to country and is used informally on social media and in everyday life [7], [11]. AD has six main regional groups: Egyptian, Levantine, Gulf, Iraqi, Sudanese (sometimes combined with Egyptian), and Maghrebi [7], [11].

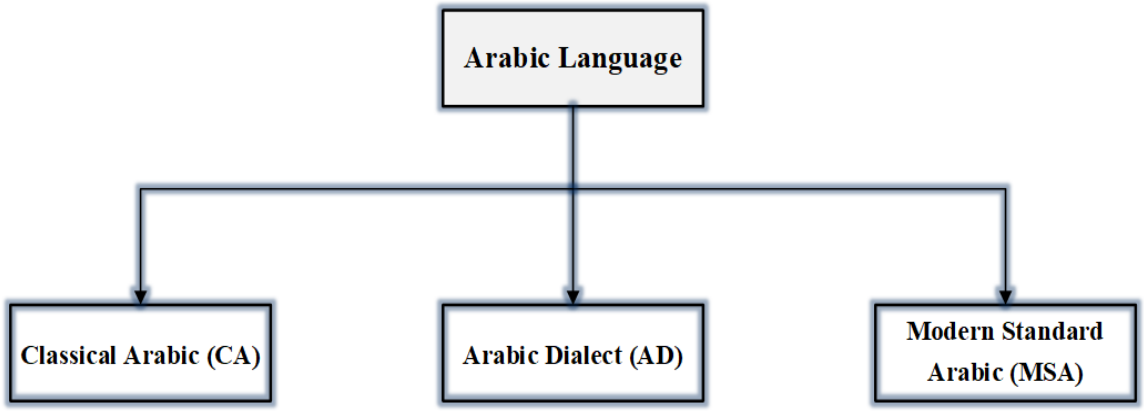

Figure 1.1.1 Arabic Language Forms

2- **Rich morphology**: One challenge facing the Arabic language is its morphological richness and complexity [12]. One Arabic word can have several meanings, either by adding suffixes, affixes, and prefixes, or through derivation, inflection, and agglutination, which leads to a high degree of complexity for ASA [9].

3- **Orthographic variations**: The Arabic language is written from right to left using 28 letters with different shapes based on their position and utilizes diacritics that may be located above or under the letters [13]. Variations in letter spelling and incorrect diacritics can alter the meaning of sentences and obscure the intended sentiment, making sentiment classification more challenging.

4- **Code-Switching:** In the informal Arabic language, users often mix MSA with various dialects and sub-dialects, as well as with other languages, especially the multiple European languages and English [14]. For example, the review “great انت شخص", which means "you are a great person". Another form of code-switching is Arabizi, which represents Arabic words with Latin characters, and is widespread among Arabic youth [9]. For example, “Allah yehfadkom” means “God save you” where “Allah” means “God” and “yehfadkom”

means “save you”. Therefore, eliminating English and other Latin characters may alter the meaning of the sentence and affect the model’s classification performance.

5- **Lack of Resources:** Large, annotated corpora that are characterized by size, dialect richness, and quality are vital to improving sentiment analysis [9]. However, such corpora are scarce and inadequate in Arabic [9]. Additionally, the lack of sentiment lexicons, which are mostly generated from existing corpora, means these lexicons often inherit the same limitations [9].

6- **Short and noisy texts**: Texts posted on social media platforms, especially Twitter, tend to be short and noisy [15]. Users often write with non-standard spelling and a lot of abbreviations, emojis, and hashtags, which complicates the task of ML models [15].

7- **Domain dependency**: Sentiment classifiers are often domain-specific, meaning a classifier trained on one domain may perform poorly when applied to another [16]. This is because the vocabulary and expressions differ across domains in sentiment analysis [17]. This problem is exacerbated in Arabic sentiment analysis due to the complexity of its morphology and the variability of its dialects, which results in feature sparsity [17].

8- **Semantic ambiguity**: Arabic words can have different meanings based on their usage in context [18]. Correctly interpreting the intended sentiment requires a deep contextual and cultural understanding of the text.

These practical and linguistic complexities create the need for specialized approaches to overcome the Arabic language challenges, considering its syntactic and semantic properties. This can be achieved by implementing the appropriate data cleaning and preprocessing strategies, such as removing duplicated rows, hashtags, numbers, and diacritics, as well as tokenization and the elimination of stop words. Furthermore, strengthening the ASA domain requires finding the right way to handle Latin words and punctuation and using suitable stemmers and transformers to return words to their correct roots. This will enable transformers to generate meaningful vector representations that can overcome Arabic morphological and orthographic challenges. Additionally, emojis are a popular feature on social media because they can enrich textual content and provide additional helpful features for sentiment analysis [19].

Emojis provide a rich source of semantic meaning that can convey users’ opinions, emotions, and feelings, especially when combined with textual data to maintain the intended meaning of tweets and social media posts. In this study, not only the emoticons expressing facial expressions were

considered, but also those used to enrich tweets and posts with thoughts, ideas, and concepts, including food and drink, weather status, animals and plants, celebrations, vehicles and buildings, and the intended emotions and feelings from their use [20]. For example, the heart emoji "❤" means "يحب شخص و الرومنسية و المودة", which means in English "loves someone, romance, and affection" while the dog emoji "🐕" means "حيوانك الأليف او الكلب و الولاء والصداقة والرفقة و الثقة", and in English means "Your pet or dog, loyalty, friendship, companionship, and trust". These emojis have a positive meaning, but other animal emojis, like the monkey emoji "🐒", may represent different sentiments. The monkey emoji means " قرد او رمز المرح والأذى والفضول او شخص شقي او عدم النضج أو الحماقة أو عدم المسؤولية او بلاغ عن شيء مريب أو مخادع أو حتى غير قانوني" and in English "monkey or symbol of fun, mischief, curiosity, a naughty person, immaturity, foolishness, irresponsibility, or reporting something suspicious, deceptive, or even illegal". On the other hand, the flower emoji "🌹" means "الحب و الرومانسية، مناسبة خاصة" and in English means "love and romance, special occasion".

The facial emoji "😀" has a positive meaning and represents "السعادة والإثارة بشكل عام", which means "happiness and excitement in general" in English. Some emojis are rich in religious emotions, such as "🕌", which means "مسجد او الإسلام و الإيمان الديني أو الهوية كمسلم او الأعياد الإسلامية او الرحلات" and in English "Mosque, Islam, religious faith, identity as a Muslim, Islamic holidays, or trips", while "🔺" is rich in meaning and intention, such as " الإعجاب. الجبال المادية أو فكرة المشي لمسافات طويلة والمغامرة بالطبيعة أو القوة أو السفر. او التغلب على التحديات، أو إحساسًا بالسلام والتأمل", which means "Physical mountains or the idea of hiking and adventure. Admiration for nature, strength, or travel. Or overcoming challenges, or a sense of peace and contemplation". Thus, eliminating such emojis could omit valuable feelings or even the real meaning of the tweet, resulting in a change to the tweet's overall meaning and emotional tone. On the other hand, combining the intended meaning and emotion of the emoji with the text of the tweet will support ML in extracting nuanced insights, creating a high-performance classification model that will support decision-makers and managers in their decision-making processes.

This thesis proposes an approach to emoji handling, introduced by creating an Emoji Meaning dataset and using it to replace each emoji with its emotional and literal meaning on social media. Furthermore, this thesis proposes a hybrid CNN-LSTM model to evaluate the impact of this preprocessing step, as well as the impact of keeping non-Arabic words, retaining punctuation, and using different stemmers and embedding transformers, on the performance and the quality of ASA,

in order to build robust prediction models. These techniques address specific ASA challenges, including the complexity of Arabic dialects, orthographic variations, the lack of resources, code-switching, and the intricacies of Arabic morphology. The proposed approach provides a more nuanced understanding of how to overcome these challenges by effectively employing these preprocessing techniques. To the best of our knowledge, this is the first work to utilize a combined CNN-LSTM approach with emoji translation for Arabic sentiment analysis, a technique that deserves consideration. An extensive literature search was conducted in multiple databases, including IEEE Xplore, Scopus, MDPI, ResearchGate, and Google Scholar, to verify the novelty of this work. The search covered the period from 2013 to 2025 and used keywords such as "Arabic sentiment analysis," "sentiment analysis," "ASA," "emoji encoding," "emoji translation," "data preprocessing," "deep learning," "hybrid model," "CNN," and "LSTM." This comprehensive search confirmed the lack of prior studies combining emoji encoding with a CNN–LSTM model for ASA.

1.2. Research Objectives

The main goal is to improve the model's understanding of Arabic text sentiments by combining emoji encoding with the hybrid CNN-LSTM model. The secondary main objective is to explore the impact of different preprocessing techniques. In other words, the secondary aim of this thesis is to investigate the importance of data preprocessing for deep learning models to achieve comparable results with other state-of-the-art models and to explore the effect of each preprocessing step. Furthermore, the techniques of cross-validation and parameter tuning are used to improve the model results. Improving ASA performance will lead to enhanced decision support capabilities in various aspects of life, including politics, tourism, e-commerce, education, and health.

Specific objectives of the study are:

1- To collect a dataset of the social media emoji meaning to explore the effect of using the emotions inside the emoji in sentiment analysis.

2- To explore the effect of keeping and removing English words and sentences from the Arabic texts.

3- Explore the effect of keeping and removing punctuation from the texts.
4- Investigate the most compatible stemmers and embedding transformers with each other and with each preprocessing step.
5- Apply the appropriate architecture for the hybrid CNN-LSTM model to be compatible with the large dataset used in this study.
6- To compare the performance metrics with other state-of-the-art studies using the same dataset.

1.3. Contributions

Although the field of sentiment analysis has advanced recently, the complexity of Arabic language processing and the lack of sentiment-specific datasets continue to hinder progress. Moreover, existing studies often fail to properly incorporate the hidden information in emojis, which are increasingly used by social media users and can complicate the proper data processing. Therefore, this thesis aims to fill this gap and contribute to the field of Arabic sentiment analysis by:

1- Adopting a hybrid model by combining a CNN model and an LSTM model to classify the sentiments extracted from the user tweets.
2- Collecting a dataset of emoji meanings (Emoji Meaning) and encoding each emoji to its real and emotional meaning. Then, combine the textual data and the textual meaning of emojis to support the hybrid model and determine the intended sentiments in the tweets.
3- Utilizing the Keras Tuner to optimize the parameters of the deep learning model, using cross-validation to ensure that the model is trained and validated on all pieces of training data.
4- Exploring whether retaining punctuation in the embedding transformation process affects tweet meaning, and the role of non-Arabic words in enriching the text meanings.
5- Exploring the ability of the trainable and pre-trained transformers and the stemmers to deal with punctuation and non-Arabic sentences, and to determine the compatibility between them.
6- Achieve high performance compared to other state-of-the-art models using different models and methods.

1.4. Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 provides a literature review of sentiment analysis and data preprocessing techniques. The first section of this chapter provides background on the Arabic language and the challenges it poses to sentiment analysis. It also suggests appropriate deep learning models for sentiment analysis. The second section investigates data preprocessing details using several datasets in conjunction with classical machine learning and deep learning models. Then, it investigates several hybrid model approaches with the datasets and transformers used. Subsequent subsections propose different strategies employed in previous studies to handle emojis and improve the SA field. These subsections also provide detailed explanations of the shortcomings and deficiencies of each study and how these challenges were addressed.

Chapter 3 outlines the proposed methodology, which includes data collection using a publicly available dataset and the development of the novel Emoji Meaning dataset. Additionally, the Exploratory Data Analysis (EDA) was proposed to provide a visual representation of the data and facilitate a clear understanding of its distribution. Next, each data preprocessing step is explained, followed by a description of the hybrid model architecture, cross-validation, and tuning processes.

Chapter 4 presents the results obtained from each experiment, reflecting the model's performance and the effect of each data preprocessing condition on the model's accuracy. A clear discussion of the results follows, along with a comparison of the proposed approach with other models that were applied to the same dataset.

Chapter 5 presents the conclusions and suggestions for future research in this area.

Chapter 2 Literature Review

2.1. Background

The Arabic language presents several challenges in sentiment analysis due to its complex morphology, diglossia, orthographic variations, and semantic ambiguity. In addition, there is a lack of resources, and social media users tend to publish short texts and use code-switching. This requires a performant model to understand the data patterns and sentiments from Arabic texts. Deep learning is a promising solution for NLP challenges because it automatically learns features that can outperform the classical ML models, where features are extracted manually [21]. CNN and LSTM models, in particular, show great performance in capturing spatial and long-term dependencies [22], [23], making them suitable for sentiment prediction in languages with rich morphology, such as Arabic.

2.1.1. Convolutional Neural Network (CNN)

CNNs are a type of Neural Network (NN) with a design that enables them to process and analyze data with a special representation [22]. They are excellent at capturing the spatial dependencies of targets and their environments, making them well-suited for several tasks, including natural language processing, image recognition, time-series prediction, and audio signal pattern recognition [23], [22]. The CNN design enables the model to learn spatial hierarchies of features automatically through backpropagation [24]. It consists of three main components: convolution layers, which consist of a stack of mathematical operations and perform the feature extraction task on the input data; pooling layers, which reduce the input dimensionality; and fully connected layers, which map the extracted features to the final output [24].

CNNs can detect the informative features in input textual data, such as word combinations and patterns, using the learnable filters in the convolutional layer to extract the features from the input data at different spatial locations [22]. CNNs are also effective in handling input of variable length due to the max-pooling layer's ability to reduce input dimensionality and generate a fixed-size output. These characteristics make CNNs suitable for the Arabic sentiment analysis field and encourage their use in the proposed model.

2.1.2. Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNNs) are a type of NN with an architecture of neurons connected with each other to form a directed cycle, called a feedback loop, within the RNN. This cycle gives the RNN internal memory, allowing it to remember the previous computations of information and pass them to the next element in the input sequence, enabling the processing of sequential information [21]. However, during back-propagation, RNNs encounter an issue where the error function explodes when there are multiple time steps [22]. Thus, LSTMs, a special type of RNN, were designed to address temporal dependencies, including time series and text sequences [22]. LSTMs address the vanishing or exploding gradient problem encountered in RNNs by incorporating a memory cell and gating mechanisms into their architecture [22]. These mechanisms regulate the flow of information through the network over time [22]. This makes LSTMs suitable for issues strongly related to the NLP field or time series analysis, as they can effectively treat sequential data with long-term dependencies [23].

The basic architecture of an LSTM unit consists of two main parts: a memory cell and three gates [22], [25]. These gates include the input gate, which is responsible for updating the memory cell with the new data, the forget gate, which is responsible for determining whether to keep or discard the data from the memory cell, and the output gate, which creates the next hidden state from the current memory cell [22], [25]. Thus, the LSTM gates are responsible for updating the current memory cell and the current hidden state [25], [26]. Consequently, the LSTM structure provides great characteristics as a strong memory maintenance over long sequences and context handling, making it well-suited for treating the complexity of the Arabic language and providing insight to use it in the proposed model.

2.2. Related Works

Sentiment analysis is the understanding of people's opinions, emotions, and attitudes toward any topic or person expressed in textual data [27]. In the field of Natural Language Processing, the ASA has recently received increasing attention [28]. Research in ASA has explored classical machine learning [29], deep learning [30], and hybrid models [31] for classifying Arabic sentiments.

2.2.1. Classical Machine Learning

Classical ML models classify feelings and sentiments based on provided features through a group of mathematical algorithms [32] and NLP techniques. The sentiment analysis task belongs to the supervised learning group. This approach works by dividing the dataset into two parts for training and testing, so that after learning the ML models on the training part, these models are evaluated by the testing part to assess the model's ability to classify the new provided data [33]. Several ML algorithms have been used for the ASA and have been previously explored to improve sentiment analysis and classification. Therefore, this section describes different studies that have used ML for ASA.

Thus, several supervised ML models have been explored in [34] to provide an Arabic sentiment analysis by applying the Support Vector Machines (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), and Random Forest (RF) models to classify customer sentiments about different coffee products. These models were trained on a dataset of 10,646 Twitter reviews after data cleaning and feature extraction using Term Frequency Inverse Document Frequency (TF-IDF) and Minimum Redundancy Maximum Relevance (MRMR) feature extraction techniques. Then, ensemble learning was used to aggregate the classification expressions and provide the final results. The best accuracy of 95.95% was achieved with hard voting and 94.51% with soft voting. Then, more ML models and feature extraction techniques were investigated in [35] by using SVM, Naive Bayes (NB), Logistic Regression (LR), KNN, DT, and RF models with 5-fold cross-validation to classify Arabic comments on YouTube and evaluate video quality before watching using a dataset of 4212 comments. Different N-gram ranges and TF-IDF methods were also explored to find the most effective features for the model. The best accuracy of 94.62% was achieved with NB using only n-grams without incorporating the TF-IDF. More techniques were evaluated in [36] with a larger dataset, integrating the RF, LR, DT, Multinomial Naive Bayes (MNB), SVM, Multi-Layer Perceptron (MLP), AdaBoost, Gradient Boosting, and Stochastic Gradient Descent (SGD) models with the resampling techniques and feature extraction methods, including TF-IDF, Bag-of-Words (BOW), and N-grams. The ML models were investigated using the ElecMorocco16 dataset, which consists of 10,254 Arabic Facebook comments written in

standard Arabic and Moroccan dialects. The best result was obtained using the SVM and TF-IDF after preprocessing and up-sampling the data with an accuracy of 90.24%.

The study by the authors of [37] investigates the SVM and NB models using the Arabic Jordanian General Tweets (AJGT) dataset of tweets written in both the MSA and Jordanian dialects. These models were used to investigate different weighting schemes, stemming, and N-gram term methods, and scenarios. The best accuracy of 88.72% was obtained when using the SVM model with TF-IDF weighting and stemming by the bigrams feature. The study by the authors of [38] used both machine learning and deep learning models for the Arabic sentiment analysis to identify the determinants of electronic service adoption spatially for the Iraqi dialect. So, the NB, KNN, SVM, RF, LSTM, Doc2vec and LSTM, and CNN were applied to the IQCD dataset, which consists of 1234 comments written by Iraqi citizens after using the data augmentation to address the imbalance in the dataset. The results show the outperformance of the machine learning model over deep learning, as the best accuracy of 82.89% was achieved by the NB model.

On the other hand, lower results were obtained by the authors of [39] by exploring six ML approaches including SVM, NB, LR, KNN, Latent Dirichlet Allocation, and Classification And Regression Trees (CART) with 10-fold cross-validation to evaluate two feature extraction techniques which are Word2Vec and TFIDF for the classification task of two and three classes. So, the classical Arabic dataset CASAD was collected from many books, resulting in 8303 paragraphs labeled by experts. The best results were achieved using the LR with word2Vec for the two-class task with an accuracy of 71.42%. In [40], by exploring the SVM, NB, Linear Regression, SGD, and Complementary Naive Bayes (CNB) with the TF-IDF transformer for both sentiment and sarcasm prediction. Based on the ArSarcasm-v2 dataset, these models were trained and tested with 5-fold cross-validation. The final results show that the SVM model achieved an accuracy of 59.8% and 74.6% for sentiment and sarcasm, respectively, and outperformed other model results.

Table 2.2.1 A summary of the studies with the best accuracy achieved using classical ML models.

Study	**Best Model**	**Dataset**	**Number of Classes**	**Text Transformer**	**Accuracy**
[34]	Ensemble model of SVM, KNN, DT, and RF	Collected about coffee products from Twitter	3	TF-IDF and MRMR	95.95%

[35]	NB	Dataset of Arabic comments on YouTube	2	n-grams	94.62%
[36]	SVM	ElecMorocco16	2	TF-IDF	90.24%
[37]	SVM	AJGT	2	TF-IDF	88.72%
[38]	NB	IQCD	3	-	82.89%
[39]	LR	CASAD	2-class task	Word2Vec	71.42 %
[40]	SVM	ArSarcasm-v2	3 for sentiment	TF-IDF	59.8%

2.2.2. Deep learning

Deep learning techniques, including Artificial Neural Network (ANN), CNN, LSTM, Bidirectional Long Short-Term Memory (BiLSTM), Deep Neural Networks (DNN), and bidirectional - Gated Recurrent Unit (BIGRU), are considered a branch of machine learning with high performance in learning the embedded and abstract data representations, which makes deep learning efficient for the sentiment analysis domain [9]. Moreover, deep learning follows the way of human learning by traversing through multiple hidden layers, so these models have a great ability to extract and transform features, which gives them more advantages for dealing with sentiment analysis [33]. Accordingly, several previous works explored the ASA using different deep-learning models and different Arabic datasets as follows.

For example, the study in [41] used the BiLSTM and CNN to evaluate several classical word embedding techniques, including Word2Vec, GloVe, and FastText, and the contextualized word embedding of the ARBERT transformer for sentiment analysis with a comparative analysis. The word embedding models were used in trained and pre-trained versions with the BiLSTM and CNN models on five datasets, including Arabic Sentiment Tweets Dataset (ASTD), ArSAS, AJGT, Khooli, and Hotel Arabic-Reviews Dataset (HARD) for ASA. The BiLSTM model outperforms CNN on the HARD, Khooli, and ArSAS datasets with accuracies of 94.25%, 84.97%, and 73.35%, respectively, using BERT-based custom embeddings, while CNN performs better on ASTD and AJGT datasets with accuracies of 49% and 80.76%, respectively, also using the trained version of the BERT transformer. Thus, the trained version transformers outperform their pre-trained

versions by about 0.28% to 1.8% in accuracy. In addition, the BERT model in both trained and pre-trained versions achieves the highest performance among all datasets. Then, based on the same HARD dataset, the study in [42] introduced two deep learning models, including the CNN and the LSTM, and the CNN-LSTM hybrid model. These models were examined with the Word2Vec and FastText feature extraction methods to find the best combination for the ASA task. The HARD and Large-Scale Arabic Book Reviews (LARB) datasets with different setups were used for training and testing the models. The best accuracy for the HARD dataset was achieved using the CNN model with FastText of 94.69% when using the 2-class imbalance dataset setup. Also, for the LARB, the LSTM model with FastText achieves the best accuracy of 86.36% when using the 2-class imbalance dataset setup.

Another study in [43] investigated the DNN along with several classical ML models, including SVM, NB, and RF. Furthermore, the Differential Evolution (DE) algorithms were used for tuning the model parameters to accurately classify the sentiment of Arabic texts related to Monkeypox. The used dataset, which consists of 4763 tweets, was collected from Twitter over eight months. The best accuracy of 92% was obtained using the DNN based on Leaky ReLU. Consequently, the study in [44] started with the creation of an ANP5 dataset, which was collected from the Arabic news post, followed by the application of semi-supervised learning with the AMCFFL technique to generate a second dataset, ANPS2. After that, the NB, RF, SVM, and LR were trained on ANP5 to achieve the best accuracy of 82% using RF and SVM among the other ML classifiers, while the DL models including BIGRU, CNN-LSTM, LSTM, and CNN were applied on the ANPS2 dataset to achieve the best accuracy of 90.10% by the CNN model. Also, the RMuBERT model outperformed BERT, ARBERT, MARBERT, and AraBERT on both ANP5 and ANPS2 datasets with an accuracy of 90.33% and 90.87%, respectively. In addition, RMuBERT was tested on several Arabic datasets, including ArSarcasm, STD, AJGT, AAQ, Books Reviews in Arabic Dataset (BRAD), and TEAD, and outperformed the AraBERT and other models with accuracies of 77.76%, 91.79%, 94.07%, 93.48%, 91.18%, and 91.12%, respectively. Also, the classical ML and deep learning were investigated in [45], which uses the SVM, KNN, NB, LR, RF, and ANN with 5-fold cross-validation to explore the Arab citizens' satisfaction with the COVID-19 mHealth apps by providing Arabic sentiment analysis of users' reviews. Thus, from the 114,499 collected reviews, a representative sample of 8220 reviews was used to train and test these models. The

findings show that the ANN algorithm with FastText embedding achieves the best accuracy of 89.1%.

A key point in [3] is that the pre-trained model is not only used to generate the embeddings. Instead, it presents a fine-tuning approach for Arabic sentiment analysis, such as the Arabic BERT pre-trained model. This approach consists of three stages, including text preprocessing and data cleaning, transfer learning of the pre-trained model weights, and the use of a fully connected layer and a dropout layer for the final classification. The model was tested on five different Arabic review datasets, followed by comparing its results with 11 state-of-the-art models to evaluate the model's performance. This model outperforms the prediction accuracy of the other models with accuracies of 93.61%, 92.21%, 96.11 %, 96.42%, and 75.17% for the ASTD, LABR, AJGT, HARD, and ArSen TD-Lev datasets, respectively. In addition, the study presented in [46] applied different versions of two transformer-based models, AraELECTRA and AraBERT, for sarcasm and sentiment classification using the ArSarcasm-v2 dataset. The best accuracy of 78.3% for sarcasm was achieved by the AraBERTv2-base model, while for the sentiment task, the AraBERTv0.2-large was the best model with an accuracy of 69.83%. While in [47] only the BERT-MINI as a transformer-based model was fine-tuned, and both the classical machine learning models, including LR, RF, KNN, NB, SVM, and XGBoost, and deep learning models, including CNN, LSTM, and BERT-MINI Arabic version, were explored to provide an ASA of tweets about the annual Hajj pilgrimage. By training and testing these models on a collected dataset of over 80K tweets about the annual Hajj pilgrimage and exploring different feature extraction techniques, including the unigram and bigram TF-IDF weighting, BOW, and Word2Vec word embedding to find the best model. The results show that BERT-MINI, when trained on the cleaned tweets without using the feature extraction methods, outperforms other models with an accuracy of 93.88%.

Table 2.2.2 A summary of the studies with the best accuracy achieved using deep learning models.

Study	**Best Model**	**Dataset**	**Number of Classes**	**Text Transformer**	**Accuracy**
[41]	BiLSTM	HARD	2		94.25%
		Khooli	3		84.97%

		ArSAS	4	BERT-based	73.35%
	CNN	ASTD	4, but they use 3 classes	custom embeddings	49%
		AJGT	2		80.76%,
[42]	CNN	HARD	2	FastText	94.69%
	LSTM	LARD	2		86.36%
[43]	DNN	Collected from Twitter	3	-	92%
[44]	RMuBERT	ANPS2	2	-	90.87%
		ANP5	5		90.33%
		AJGT	2		94.07%
		AAQ	2		93.48%
		STD	2		91.79%
		BRAD	2		91.18%
		TEAD	3		91.12%
		ArSarcasm	3		77.76%
[45]	ANN	Arab citizens' satisfaction with the COVID-19 mHealth apps	3	FastText	89.1%
[3]	Arabic BERT	HARD	2		96.42%
		AJGT	2		96.11%
		ASTD	4, but they used 2		93.61%
		LABR	2		92.21%
		ArSen TD-Lev	3		75.17%
[46]	AraBERTv0.2-large	ArSarcasm-v2	3 for sentiment	-	69.83%

[47]	BERT-MINI	Dataset of the annual Hajj pilgrimage	3, but the model was trained on 2 classes	-	93.88%

2.2.3. Hybrid models

Hybrid deep learning models integrate multiple deep learning architectures to leverage their strengths and enhance model performance and robustness. Combining different deep learning models can lead to overcoming the limitations of single models, such as contextual meaning, semantic dependency, and related challenging issues, and improve the accuracy of sentiment analysis [48]. In this section, various studies have been reviewed that have used hybrid deep learning models to overcome the complexity of Arabic sentiment by exploring these combinations on different Arabic datasets.

One such combination was presented in a study [49], which introduced a hybrid AraBERT-LSTM model in which the output of the AraBERT model is fed into an LSTM model. Feedforward neural networks and an output layer follow this. Several ML and deep learning models were also examined to find the best model for enhancing the ASA field, including RF, LR, DT, SVM, NB, KNN, LSTM, and Gated Recurrent Unit (GRU). Only the ISRI stemmer was used for stemming, while several feature extraction techniques were explored: AraBERTv0.2 for the hybrid model, TF-IDF for the machine learning model, and Word2Vec's Continuous Bag-of-Words (CBOW) and Skip-Gram architectures for the deep learning model. This approach was applied to the Arabic Sentiment Analysis Dataset (SS2030) [50], Arabic Sentiment Twitter Corpus (ASTC) [51], Arabic Health Services Dataset (Main-AHS and Sub-AHS) [52] datasets to train and test the model. The AraBERT-LSTM model achieved the best testing results for all datasets, with respective accuracies of 90.40%, 93.76%, 92.61%, and 97.12%. Another combination was proposed in [53], which applied an ensemble model combining CNN and LSTM on the AraSenTi [54] dataset, consisting of 15k, collected from Twitter. Furthermore, the word and character level embeddings were explored for the hybrid model. Then, two other deep learning models were explored, including CNN with the character level embedding and LSTM with the word level embedding

using FastText, by training and testing these models on the same dataset. Also, the hyperparameter tuning with 10-fold cross-validation was applied to optimize each model. The final result shows that the ensemble CNN-LSTM model achieved the best accuracy of 96.7% using 10-fold cross-validation. A different model combination was used in [55] by merging the RNN with the BiLSTM to generate the RNN-BiLSTM hybrid model. Then, the embeddings generated by the AraBERT transformation model from the Arabic Reviews Dataset (ARD) [56], which consists of 100k opinions, were used to train and test the model; the best accuracy achieved was 95.75% with the default parameters. Moreover, the RNN-BiLSTM model was tested on two other datasets to explore the effect of transfer learning, including the ASTD [57], which consists of 10k tweets, and the Gold Standard Corpus (GSC) AraCust dataset [58], which consists of 20k customer reviews, after fine-tuning the model parameters using random search it achieves accuracies of 95.44% and 96.19%, respectively. Then, other feature extraction techniques used in [59], including FastText and GLOVE, were used to generate embeddings to be used with the explored models. So, both the deep learning models and hybrid models, including CNN, LSTM, GRU, BiGRU, BiLSTM, CNN-BiGRU, CNN-GRU, CNN-LSTM, and CNN-biLSTM, were used for predicting the Arabic sentiment on two datasets, which are the HARD, consisting of 93.7k comments, and BRAD, containing 510.6k book reviews. The best accuracies of 87.46% and 91.12% were obtained for the HARD and BRAD datasets, respectively, using the CNN-biLSTM with FastText embedding.

In addition, the study in [60] integrates the CNN and LSTM to classify sentiments in the Arabic tweets and compares the model results with the deep learning models, including the CNN and LSTM individually, to evaluate the model performance. The dataset used was collected from Facebook discussions in 2016, resulting in 10254 Arabic comments written in Moroccan dialect (Darija) and modern standard Arabic. After preprocessing the data, the FastText library represents each word as a bag of character n-grams and generates word embeddings to be used for training and testing CNN, LSTM, and CNN-LSTM models. After evaluating the models, the CNN-LSTM model outperforms the individual models and provides real improvements in sentiment classification accuracy. Consequently, another combination was introduced in [61] by combining the CNN with LSTM models to be applied to three datasets, which are ASTD, Main-AHS and Sub-AHS, and Ar-Twitter [62] datasets. Then, in order to maintain the same length of the feature vector after the filters convolve on the input data, the max-pooling layer was excluded from the CNN. In addition, Farasa, MADAMIRA, and Stanford were used as data preparation techniques

for preprocessing the Arabic text, and several pre-trained transformational models were investigated to generate vector representations for the text, including Glove, FastText, and Word2Vec, and to improve the Arabic sentiment classification accuracy. The best accuracy of 94.83% for the Main-AHS dataset was obtained using Farasa Lemmatization normalization, 88.86% using Madamira Stem normalization for the Ar-Twitter dataset, and 81.62% using Word2VecSG embeddings for the ASTD dataset. Subsequently, the study in [63] proposed a more complex model that combines the AraBERT contextualized sentence representations with the pre-trained Mazajak static word embeddings. Thus, the CNN-BiLSTM model generates sentence representations from the Mazajak static word vectors in order to concatenate them with the AraBERT embeddings, resulting in the final sentence representation. The CNN-BiLSTM model outperforms the standalone AraBERT model using the ArSarcasm-v2 [64] dataset for both sarcasm and sentiment classification tasks. The best accuracies are 86% and 75% for sarcasm and sentiment classification, respectively. Also, the CNN-BiLSTM was used in [65] as a multi-task model including a topic classifier, a sarcasm detector, a sentiment analyzer, and an emotion classifier by training the model on different datasets for each task. So, for sentiment analysis tasks, four datasets were used, which are the Twitter Dataset for Arabic Sentiment Analysis, ArSAS, ArSarcasm-v2, and SS2030 datasets, which consist of 348,797, 21,000, 15,548, and 4214 tweets, respectively. The accuracies achieved by the proposed model were 97.58%, 97%, 86%, and 81.6% for the topic, sarcasm, sentiment, and emotion classification, respectively.

Table 2.2.3 A summary of the studies with the best accuracy achieved using hybrid models.

Study	**Best Model**	**Dataset**	**Number of Classes**	**Text Transformer**	**Accuracy**
[49]	AraBERT-LSTM	SS2030	2	AraBERTv0.2	90.4%
		ASTC	2		93.76%
		Main-AHS	2		92.61%
		Sub-AHS	2		97.12%
[53]	CNN-LSTM	AraSenTi	4	FastText Word-level and	96.7%

				character-level embedding	
[55]	RNN-BiLSTM	ARD	2	AraBERT	95.75%
		ASTD	4, but they used 2 classes		95.44%
		GSC AraCust	2		96.19%
[59]	CNN-biLSTM	HARD	2	FastText	87.46%
		BRAD	2		91.12%
[60]	CNN-LSTM	Collected from Facebook discussions	2	FastText	-
[61]	CNN-LSTM	Main-AHS	2	-	94.83%
		Ar-Twitter	2	-	88.86%
		ASTD	4, but they used 2 classes	Word2Vec SG	81.62%
[63]	CNN-BiLSTM	ArSarcasm-v2	3	Combines AraBERT and Mazajak	75% for sentiment classification
[65]	CNN-BiLSTM	Merge the Twitter Dataset, ArSAS, ArSarcasm-v2, and SS2030	3	Word2Vec	86%

2.2.4. Emoji Handling Techniques

Emojis handling and treatment follow different strategies by researchers to develop SA performance, such as just eliminating the emojis, while other researchers have considered the emojis' significance in their works [66]. Retaining the emojis can help to express the writers'

feelings in the texts, which helps to improve the sentiment prediction and classification performance [67], while eliminating them can hide the rich feelings and meanings desired from the text.

Some studies exploit the emoji in SA using the strategy of replacing emoji with textual meaning, such as translating emoji in [68] by performing Unicode translation of emoji. Also, the effect of combining LSTM, RNN, and GRU in conjunction with RF, LR, and SVM implemented with CBOW word embedding while using the grid search technique was investigated to improve the Arabic sentiment analysis performance. The results are compared with LSTM, RNN, and GRU as deep learning models implemented with Twitter-CBOW word embedding and tuned using Keras-tuner, and with DT, KNN, LR, RF, and NB as ML models implemented with TF-IDF for feature extraction and tuned using grid-search cross-validation. Then, three datasets, including ArTwitter [69], ASTC, and AJGT, are used for model training and testing. The highest testing accuracies of 93.4% and 92.22% were achieved using Stacking LR of the AJGT and ASTC datasets, respectively, while the highest accuracy of 92.24% was achieved using Stacking SVM for the ArTwitter dataset. Also, in the study of [70], a RuSentiment dataset [71] of 6957 posts written in Russian and containing at least one emotional indicator (emoji, punctuation marks that express emotions, emoticons) was used. Then, each emotional indicator was replaced with its textual meaning to improve the performance. The best results were achieved by an ensemble model of the Word2Vector model and an emotional indicator model, with an accuracy of 91%. In addition, the study in [72] used the BiLSTM model with the AraVec word embedding to classify emotions using the SemEval 2018 task1, the E-C subtask dataset [73] for the Arabic language with 4381 tweets. The developed model was applied with and without translating the emoji for their related meanings to emotions based on the manually created lexicon, by transcribing each emoji to its Arabic meaning. The best results were obtained when using the emoji lexicon, with an accuracy of 49.8%. Another strategy for using the emoji to improve SA is also to use them as non-verbal features. Such as the study in [66], adapted non-verbal features for the task of Arabic sentiment analysis by investigating and evaluating several ML models, including NB, MNB, DT, Sequential Minimal Optimization-based Support Vector Machines (SMO-SVM), SGD, and RF, using emoji-based features with a feature vector of length 429 and for 2091 instances. The highest accuracy of 80.34% was achieved using the MNB model when applied to the top 250 most relevant emoji selected by ReliefF, which computes a weight for each emoji, and Correlation-Attribute Evaluator (CAE)

feature selection techniques, which find the Pearson's correlation between the emoji and the polarity label. Then, in [74], the non-verbal features were evaluated using several ML models, including KNN, Gaussian NB, SGD, SVM, DT, Bi-LSTM, LSTM, GRU, and BiGRU. Thus, after excluding tweets without emoji, a dataset of 2091 microblogs was collected from Syria [75], ArTwitter, QCRI, ASTD, Semeval-2017 Task4 Subtask#A, and 843 Arabic microblogs with emoji from Twitter and YouTube. Then, an emoji lexicon called the Emoji Sentiment Ranking (ESR) lexicon of 969 emojis was used after excluding the unused emojis, and also the Principal Component Analysis (PCA) was applied to reduce the dimensionality of the features from 430 to 100 features. The BiGRU model achieved the best accuracy of 71.71%.

The third strategy used in [76], since the textual features combined with the emoji weights as a non-verbal feature by creating an Emoji Sentiment Lexicon (Emo-SL) specifically designed for Arabic tweets from 58k Arabic tweets by computing the sentiment scores of 222 emojis commonly used in the datasets. Then, the SVM, NB, RF, and KNN are trained and evaluated using the emoji weights combined with text-based feature extraction through lexicon, since it improves the model accuracy by 26.7%. The KNN model achieved the best accuracy of 88.7 % when combining the Emo-SL with ML for the Arabic tweets and emojis. Another study in [19] used both the non-verbal and textual features to improve the model's performance. Thus, five datasets were used after removing the observations without emoji, including Syria, ASTD, ArTwitter, QCRI [77], and Semeval-2017. Based on the merged datasets, each tweet was divided into textual and emoji features. For feature extraction from texts, the Latent Semantic Analysis (LSA), TF-IDF, and Word2Vec word embedding with both architecture CBOW and skip-grams were used for extracting textual features, which were then concatenated with the non-verbal features obtained using a set of 120 emojis to determine the occurrence of each emoji. The best accuracy of 83.02% was obtained using the SVM by fusing skip-gram features with emoji and using correlation-based feature selection.

While in [78], a different approach was applied using word sense embedding generated by bi-sense emoji embedding concatenated with the text embeddings, followed by training an attention-based long short-term memory network on the generated embeddings. The bi-sense emoji embedding generates a sentiment-aware embedding of emoji since the bi-sense emoji were learned individually under negative and positive sentimental tweets. The Multi-Level Attention-based Bidirectional LSTM with bi-sense emoji embedding (MATTBiE-LSTM) model achieves the best

accuracy of 90% on the AA-sentiment dataset with bi-sense Emoji Embedding concatenated with LSTM unit encoder embedding, and the Word-guide Attention-based LSTM with Bi-sense Emoji Embedding (WATT-BiE-LSTM) achieves an accuracy of 83.4% on the HA-sentiment dataset with Bi-sense Emoji Embedding concatenated with FastText embedding. On the other hand, the study in [79] leverages deep learning for the sentiment analysis in Thai hotel reviews using a dataset of 12450 reviews containing both text and emoji. By following the strategy of integrating the CNN for emoji classification with the RNN model for the text classification task and combining them using the stacking model framework, the model results in an accuracy of 92.4%, which outperforms other models.

With the great development of Large Language Models (LLMs), the study in [80] uses the LLMs to transform the Arabic text into an embedding depending on the LLM's ability to deal with the emoji, including Universal Sentence Encoder (USE), XLM-RoBERTa base (XLM-R base), and MiniLM-L12-v2. Thus, the ASTC dataset with 56795 tweets was used to train and test the LR, SVM, RF, KNN, and DT models with and without the emojis. These LLM transformers can handle the emojis and provide a numerical representation of them to be fed to the ML models. The best results were obtained by including the emojis and using the RF model with the MiniLM embeddings, with an accuracy of 85.98%.

Table 2.2.4 A summary of the studies that follow an emoji-handling strategy.

Study	**Best Model**	**Dataset**	**Number of Classes**	**Text Transformer**	**Emoji Treatment**	**Accuracy**
[68]	Stacking LR	AJGT	2	CBOW	Unicode translation	93.4%
		ASTC	2			92.22%
	Stacking SVM	ArTwitter	2			92.24%
[70]	An ensemble model of Word2Vector and a model of the emotional indicator	RuSentiment	5, but 2 classes were used	Word2Vector	Emoji replaced with text	91%

[72]	BiLSTM	SemEval 2018 task1, the E-C	12	AraVec	emoji translating based on a manually created lexicon	49.8%
[66]	MNB	Arabic dataset of 2091 instances	2	-	emojis lexicon with ReliefF and CAE	80.34%
[74]	bidirectional - GRU	2091 rows collected from Syria, ArTwitter, QCRI, ASTD, Semeval-2017 Task4 Subtask#A, and 843 Arabic microblogs from Twitter and YouTube	2	-	Emoji Sentiment Ranking (ESR) lexicon with PCA	71.71%
[76]	KNN	58k Arabic tweets	2	textual sentiment lexicon	Weights of emojis using the Emo-SL lexicon	88.7%
[19]	SVM	Used Syria, ASTD,	2	Word2Vec skip-gram	Emoji occurrence	83.02%

		ArTwitter, QCRI [77], and Semeval-2017 Task4 Subtask#A				
[78]	MATTBiE-LSTM	AA-sentiment	2	LSTM Embedding layer	bi-sense emoji embedding	90%
	WATT-BiE-LSTM	HA-sentiment	2	FastText		83.4%
[79]	Stacking model with CNN-RNN as the base model	Dataset of Thai hotel reviews	3	TF-IDF	CNN for emoji classification	92.4%
[80]	RF	ASTC	2	MiniLM	MiniLM	85.98%

2.2.5. Knowledge Gaps

Previous studies have used various classical ML, deep learning, and hybrid models; transformers; and emoji-handling strategies for ASA and achieved promising results. However, further investigation is needed on sentiment analysis for the Arabic language due to its morphological complexity, as well as the effect of non-Arabic words, punctuation, and emojis on the meaning of Arabic text and the sentiment it conveys.

Although the studies in [53], [55], [59], [60], [61], [63], [65] applied different combinations of deep learning models to perform hybrid models on several datasets, they did not explore these models on the ASTC dataset. Additionally, none of the aforementioned studies or study [49] examined the impact of punctuation, emoji meaning, and sentences in other languages on the final classification performance, particularly with Keras and AraVec transformers. The studies in [41], [42], [43], [44], [47], [45], [3], and [46] investigated different trained and pre-trained models with various combinations of transformers. However, they did not examine the impact of the data

preprocessing on these models, particularly the influence of emoji and Keras and AraVec transformers on the model outcomes. Other studies applied the classical ML techniques, such as those in [40] and [39]. However, these models could not overcome the complexity of the Arabic language and therefore did not achieve high accuracy. Conversely, good results were achieved in [34], [35], [36], [37], and [38], which investigated several ML models and datasets. However, the characteristics of these datasets are not huge, which helped these simple models obtain good results. Furthermore, the effect of different data preprocessing was not investigated during these experiments.

On the other hand, the studies in [68], [70], [72] treated the emojis by replacing them with textual data, then applied stacking, ensemble, and deep learning models with several transformers. These studies did not explore the hybrid models or the compatibility between stemmers, transformers, and other preprocessing steps while translating emojis. The models in [68] and [70] performed well in sentiment classification, whereas the deep learning model in [72] had very low accuracy. In contrast, the studies in [66] and [74] applied the models based on the emoji features as a non-verbal variable and excluded texts that were rich in meaning and emotion, which could inspire the models. This explains the moderate level of accuracy of these techniques.

So, in [76], [19], [78], both the non-verbal features such as weight, and the original text were concatenated using different approaches to leverage the sentiments in both, resulting in higher accuracy rates than the studies that excluded text. A similar approach was used in [79], where a stacking model with a CNN-RNN as the base model stacked the TF-IDF embedding with the CNN emoji classification to achieve high classification accuracy. However, [80] depends on the LLM model for text, emoji, and data preprocessing without studying their effects. Previous studies have handled emojis with different strategies. Nevertheless, the effect of removing or keeping non-Arabic words and punctuations, choosing the best transformers when words are written in other languages alongside Arabic tweets, and translating emojis to their real and emotional meanings on classification performance has not been investigated.

This study trained and tested a CNN-LSTM on the ASTC dataset to improve ASA performance. It also examined how various data preprocessing experiments and conditions impacted the hybrid model. This examination aimed to determine the importance of each data preprocessing step, including emoji encoding, keeping or removing punctuation and non-Arabic words, using Keras

and pre-trained AraVec transformers, and assessing the compatibility of those transformers with the other preprocessing steps, with respect to model performance.

Chapter 3 Methodology

3.1. Proposed Method

This chapter proposed an overall methodology used to overcome the ASA challenges by combining the preprocessing conditions and deep learning. The developed architecture is divided into three main stages, beginning with the critical data collection stage necessary for training the ML models. This stage involved using a publicly available dataset called ASTC, as well as the Emoji Meaning dataset, which we collected as part of this research effort to enrich the ASTC dataset with the emojis' real and emotional meanings.

The second stage is data preprocessing, which is an important part of developing accurate and realistic classification models. This stage reduces data dimensionality and complexity by removing unnecessary text, numbers, symbols, and characters. At the same time, data preprocessing can enrich the social media posts with the users' intended feelings by translating emojis into their meanings to improve performance. Therefore, the fundamental goal of this research is to investigate the effect of data preprocessing on the ASA field. This stage included fixed steps, starting with dropping duplicates, followed by removing hashtags, numbers, diacritics, and stopwords, as well as tokenization. There were also variable steps to form three experiments examining the effects of non-Arabic words, punctuations, and eight conditions specialized to examine the effect of emoji encoding, stemmers, and transformers.

The third stage involves building, training, and testing the CNN-LSTM model. Cross-validation was also employed to examine how well the CNN-LSTM model generalizes to unseen data, and the Keras Tuner was used to optimize the hyperparameters during the model training phase. Furthermore, quantitative methods will be used to evaluate the model's performance on testing data and its ability to handle unseen data during the model testing step. Figure 3.1.1 shows the overall framework used to achieve the goal of this thesis.

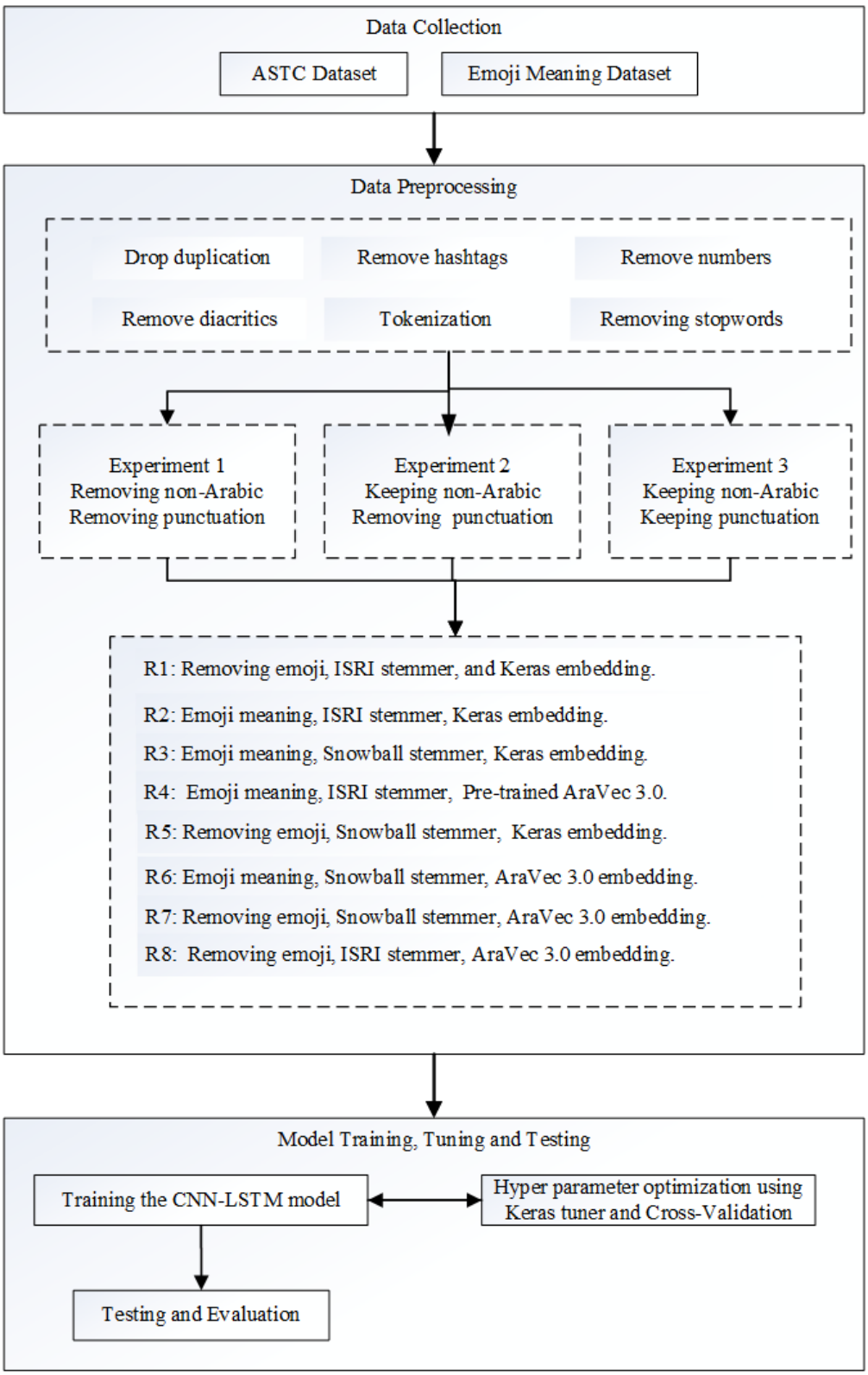

Figure 3.1.1 The workflow of the proposed model for Arabic sentiment analysis.

3.2. Dataset Description

This section proposes two datasets, namely the ASTC and the Emoji Meaning dataset. These datasets are used to evaluate the model performance and the effect of data preprocessing on the model quality. The ASTC dataset consists of labeled Arabic tweets with positive and negative sentiment polarity for training and evaluating the model, while the Emoji Meaning dataset is a dictionary of emoji meanings. This dictionary is used to map each emoji in the ASTC dataset to its meaning in order to determine the effect of emoji encoding on improving ASA performance.

3.2.1. Arabic Sentiment Twitter Corpus Dataset

The Arabic Sentiment Twitter Corpus dataset [51] is a publicly available dataset on Kaggle. It was collected in April 2019 using a positive and negative emoji lexicon, resulting in a target variable that describes the sentiment of tweets with positive and negative labels. The ASTC dataset, as described in [51], consists of 58k Arabic tweets, with 47k for training and 11k for testing. However, it actually consists of fewer than 57k labeled Arabic tweets. The dataset is divided into 80% for training and 20 % for testing, resulting in 45k tweets for training the model, with 22,760 positive and 22,513 negative tweets, and 11k for testing the model, with 5751 positive and 5767 negative tweets, to create a balanced dataset according to the target variable for both the training and testing parts.

3.2.2. Emoji Meaning dataset

The Emoji Meaning is a novel dataset, which consists of 912 emojis collected from the ASTC dataset, and was created during this work. The Emoji Meaning dataset was used to map each emoji in the ASTC dataset to its possible meaning. Some emojis have an emotional meaning based on how Twitter users employ them in their posts. For example, "☺", this emoji means " السعادة بشكل عام أو الود او عندما يكون الشخص ساخر أو عدواني بشكل سلبي" and in English means "general happiness or friendliness or when someone is being sarcastic or passive-aggressive" and “😱”, which means “الذعر أو الخوف أو الصدمة العنيفة” in Arabic and “panic, fear, or violent shock” in English. Other emojis may have only a real meaning, such as “✓”, which means “شيء صحيح" and in English "something right", and "💼”, which means “الأعمال المصرفية والتسوق والمعاملات المالية او الإنفاق والديون” and in English

“banking, shopping, financial transactions, or spending and debt”. Some emoji have both emotional and real meanings. Such as "🐕", which means " حيوانك الأليف او الكلب او الولاء والصداقة والرفقة و الثقة" and in English "your pet or dog or loyalty, friendship, companionship, trust". Also, “🍵”, which means “ لشاي، أو المشروبات الساخنة او الاسترخاء والراحة والصحة او السلام أو الهدوء أو التأمل او مشاركة النميمة أو الأخبار” and in English means “for tea, hot drinks, relaxation, comfort, health, peace, quiet, meditation, sharing gossip or news”.

So, to find all the emoji meanings based on their use by Twitter users, four specialized websites [81], [82], [83], [84] in collecting social media emoji and their meanings were used to translate each emoji to its meaning and create the Emoji Meaning dataset. The consistency of the emoji meanings was validated using these websites, which provide cross-referencing with several emoji interpretation databases or dictionaries. Additionally, a strategy was employed to handle ambiguity and validate emoji meaning since emojis can have multiple meanings by combining the commonly shared emotional and real-word meanings of emoji among Arabic-speaking users, provided by references websites [81], [82], and English-speaking users, provided by references [83], [84], and supported by human judgment. The final step in validating the emoji translation process and dataset creation was comparing model performance on the ASTC dataset with and without emoji translation, demonstrating that including emoji encoding improves the model’s robustness.

The Emoji Meaning dataset, a rich source of emotion and context, improves model performance on the ASTC dataset by replacing each emoji with its emotional and real-world meaning from the Emoji Meaning dataset. This thesis focuses on utilizing and interpreting emojis according to their global cultural norms due to the fact that emoji interpretations may differ across cultures, forming a consistency issue. Nevertheless, further investigation is still required to identify common and uncommon emojis that may have several meanings depending on the user’s culture or social media platform.

3.3. Exploratory Data Analysis (EDA)

Data visualization is described as the process of presenting data in an effective and clear pictorial or graphical way, making insights easy to understand [85]. It is also an important step before applying any machine learning or deep learning model [86]. This step helps understand the nature of the data and extract relationships between the features and their hidden patterns [86], a process

known as exploratory data analysis. Data visualization techniques are also used for other purposes, like presenting the results in graphical format and tuning the machine learning models, as discussed in [87]. In this section, data visualization techniques were employed to explore the dataset distribution and the character of the data.

3.3.1. ASTC Data Distribution

In this manner, the par plot proposes the number of tweets in each class in a visual way that simply shows the balance of the ASTC dataset, as shown in Figure 3.3.1. The distribution analysis shows sampling techniques are unnecessary and that a fair and strong ML model can be developed since the target variable is balanced with positive and negative sentiments.

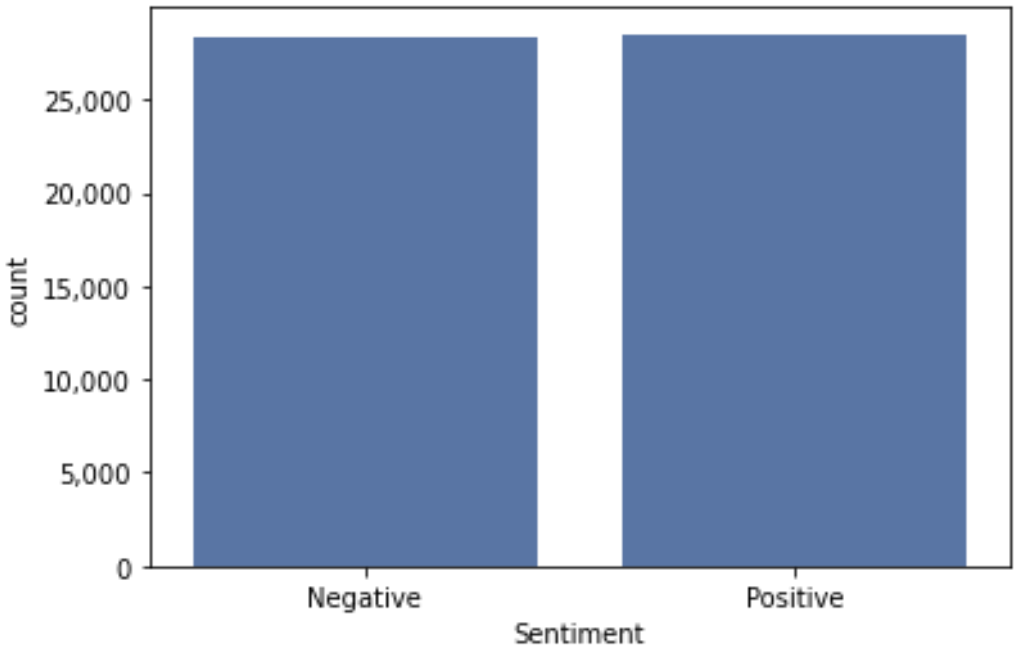

Figure 3.3.1 The number of positive and negative tweets.

3.3.2. The Character of the Data

The character of the data is understood using word cloud techniques, which will propose the most frequent words in the dataset. These techniques also demonstrate the effect of emoji encoding on enriching tweets with emotions and sentiments. Thus, before encoding the emojis, a word cloud

was generated for the ASTC dataset, as shown in Figure 3.3.2, to understand the nature of the texts in the tweets.

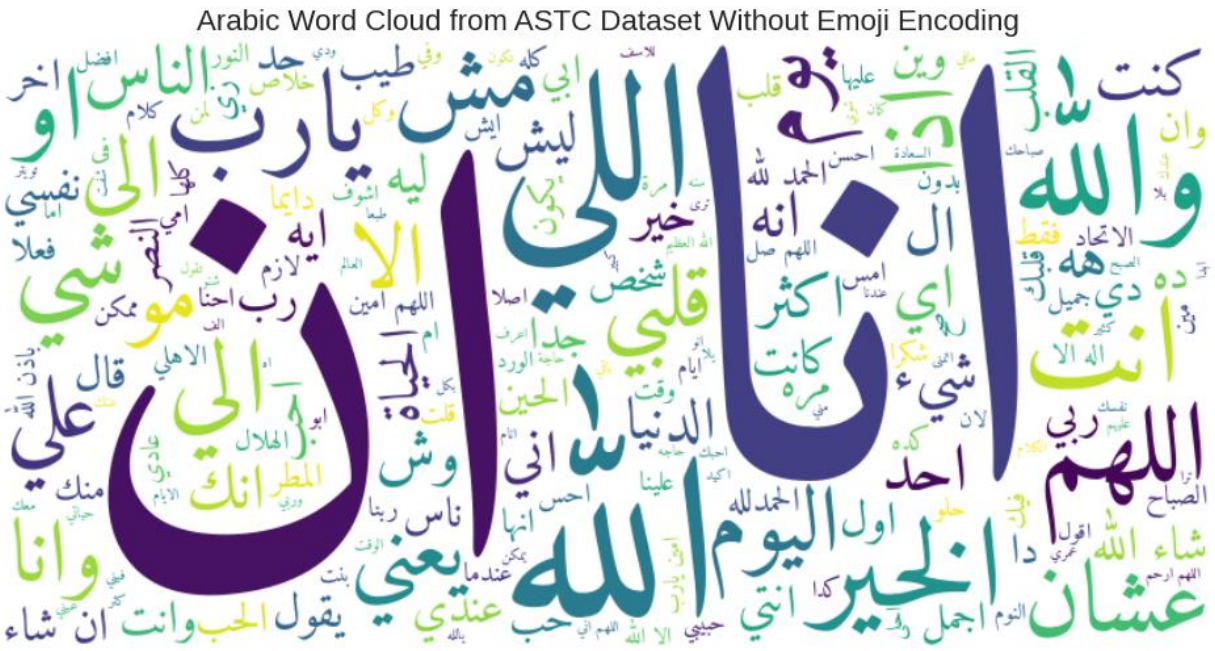

Figure 3.3.2 Arabic Word Cloud of ASTC Dataset Without Emoji Encoding.

Figure 3.3.2 shows a word cloud of the most widely used words in the dataset, such as the word "انا", which means "me", and "ان" ,which is a condition that means "if". It also contains meaningful words such as "الله", which means "God", "القلب" means "heart", "الخير" means "goodness", and the supplication "يا رب", which means "O Lord", which convey positive emotions. Other positive, albeit less frequent, words include "الحب" which means "love", "النصر" which means "the victory", "جميل" which means "beautiful", and "بإذن الله" which means "God willing" that express the love and hopeful feelings. Conversely, only the word "للأسف" meaning "unfortunately" is presented in the word cloud with negative connotations. This creates an initial impression of emotional imbalance in the data, which affects the performance of any model before it is trained.

Figure 3.3.3 shows a word cloud of the Emoji Meaning dataset to illustrate the nature of words representing emoji meanings and the distribution of positive and negative words.

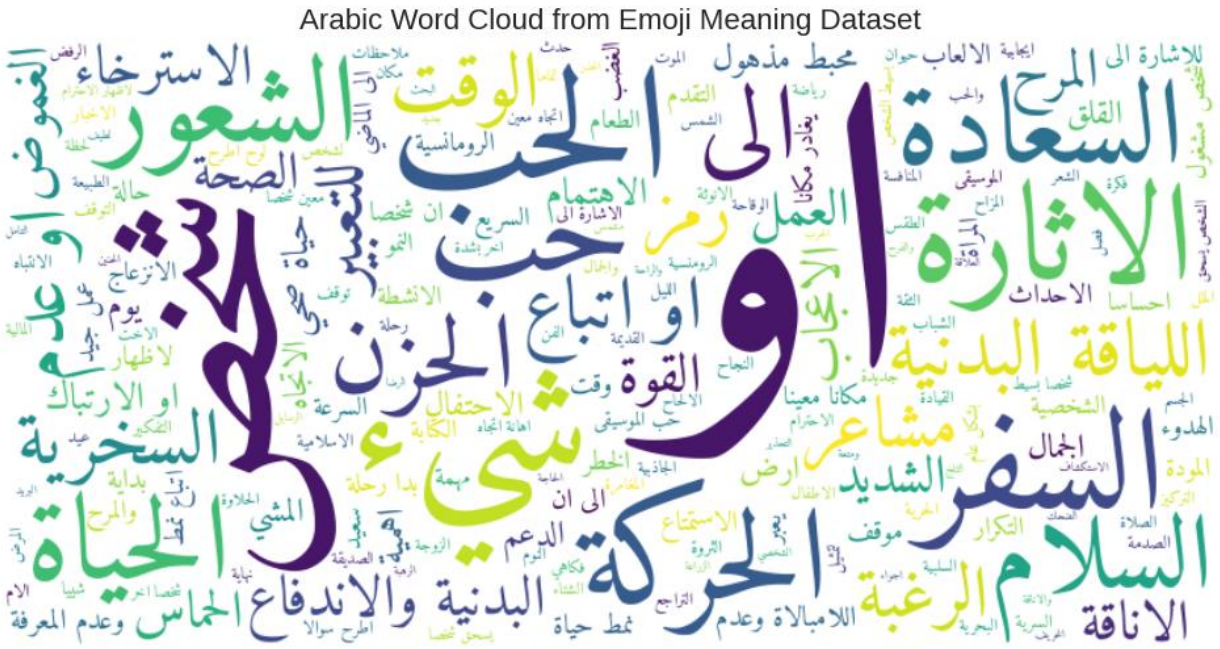

Figure 3.3.3 Arabic Word Cloud of Emoji Meaning Dataset.

Consequently, the word cloud in Figure 3.3.3 shows a set of words that are more closely related to emotions. Some of these words tend to represent the feelings of comfort, courage, and relaxation, such as the word “السعادة", which means "happiness". Other words include “الاثارة", which means "excitement", "اللياقة البدنية” means “Physical fitness”, and “السلام” means “peace”. While other words cover negative feelings, such as “السخرية", which means "Sarcasm", "الغموض” means “ambiguity”, “اللامبالاة” means “indifference”, “الارتباك” means “confusion”, and “عدم المعرفة” means “lack of knowledge”. Inspired by the Emoji Meaning dataset, this group of words can enrich other social media datasets with meanings and feelings that can enhance the model's understanding of users’ sentiments in tweets.

After translating each emoji in ASTC to its meaning from the Emoji Meaning dataset, the word cloud in Figure 3.3.4 was generated to demonstrate the effect of using emojis to enrich the dataset with emotions and feelings.

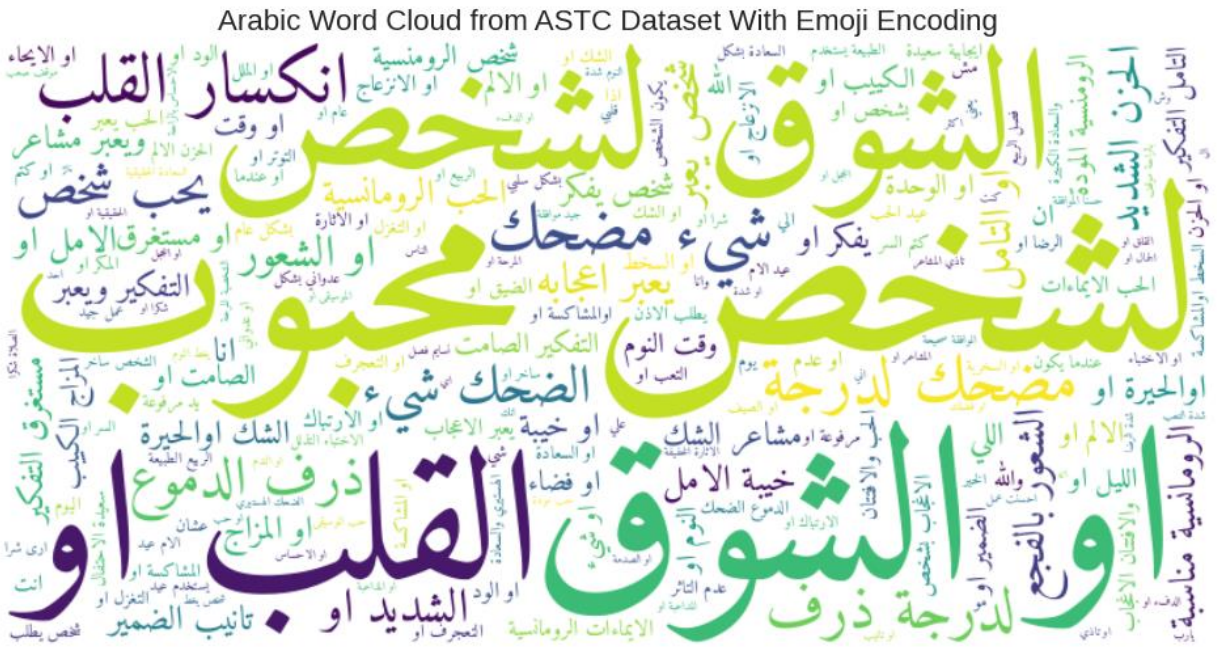

Figure 3.3.4 Arabic Word Cloud of ASTC Dataset With Emoji Encoding.

The resulting dataset proposed an impressive combination of meaningful, sentiment-rich words, as reflected by the word cloud in Figure 3.3.4. This combination presents a set of words with positive connotation, such as the words in "الشوق لشخص", which means "longing for someone" in English. Other examples include "شخص محبوب", which means "beloved person", "القلب" means "the heart", and "شخص رومنسي" means "romantic person". There are also less frequent positive words, such as "الافتتان" which means "infatuation", and others that indicate happy occasions, such as " عيد الام" which means "Mother's Day". On the other hand, after translating the emojis, the ASTC dataset became more balanced, as the number of words that reflect negative sentiments increased. Examples include the words "انكسار القلب" meaning "heartbreak", "الحزن الشديد" meaning "extreme sadness", "الشعوربالفجع" means "feeling of grief", and "خيبة" means "disappointment". Before translating the emojis, the word cloud in Figure 3.3.2 shows that the dataset lacks words with negative connotations.

3.4. Data Preprocessing

The ASTC dataset contains many duplicate rows, numbers, hashtags, stop words, punctuation, non-Arabic words, symbols, and diacritics. Therefore, we used several data preprocessing techniques to address these issues and clean the text. We followed the best practices for dealing with these issues and removing the tokens that did not contribute to the actual text's meaning. This also helped reduce the text's dimensionality. The following data preparation steps were used to evaluate our proposed model:

1- Drop duplication: Identical data repeated multiple times within a dataset may affect the accuracy and reliability of the subsequent analysis results, making meticulous cleaning important [88]. Thus, we dropped 18399 rows from the ASTC, which consists of 56k rows divided into training and testing parts. The training part contains 15721 duplicate rows, while the testing dataset contains 2678 duplicate rows.

2- Remove hashtags: Hashtags play a critical role in social media. On the Twitter platform, users widely use the hashtag character "#" to bookmark their tweets' content or follow a topic or trend community [89], [90]. The hashtag character provides important information to users who don't know the real author, signaling the author's intent [91]. Therefore, in this study, we removed the hashtag character while keeping the word following it to help the ML model understand the author's intent. For example, ". لآ تغرگ ضحگتي وتقول❌ ركز بالضحگهہ❌ تسمع صوت... #الونين ;ركز بالضحگهہ #مرتاحہ☺ ☹" which means "Don't let my laughter fool you, you say: #I'm_comfortable. Focus on laughing, and you will hear the sound of... #groans." This results in the elimination of the hashtag character while keeping the words "مرتاح" meaning "I'm comfortable", and "الونين" meaning "groans", both of which words play a role in forming the sentence's meaning.

3- Remove diacritics: Arabic diacritics play an important role in indicating short vowels, which lead to the correct pronunciation of words. The presence or absence of these diacritics can change the meaning of the words [92]. Moreover, in the world of social media, diacritics are used to give words and sentences an aesthetic shape without following the rules of the Arabic language. For example, the Fatha of " ً " is used after the broken heart emoji " 💔 ً " in "سمعت بكاه نهار العزا من بعد قفوا معزينه💔 ليا" which means in English "I heard him crying on the day of the funeral, then they stood up to offer their condolences to

me 💔 ً ". Thus, we eliminated the diacritics from the sentences in this study because they did not affect the SA measurements [93].

4- Remove numbers: we removed all numbers from tweets because they are useless and don't contribute additional information or reflect the emotions contained in the text [94], [95]. For example, the number in "12:ص لم تكن دمعات ، بل لهيب فاض من بركان 🗣" represents the time of the tweet but has no effect on the sentiment, which means "12: AM it wasn't tears, it was flames that erupted from a volcano 🗣".

5- Removing stop words: Stop words are common and frequently used in the Arabic and English languages [93] with little semantic value [96]. Removing stop words includes eliminating words that occur frequently but have little meaning, which may diminish the impact of frequency differences among less common words [97]. This reduces the computational cost and input dimensionality. In this thesis, we removed only the Arabic stop words when only Arabic words were kept in the dataset after removing words written in other languages, as well as both Arabic and English stop words if both languages were kept.

6- Tokenization: Tokenization involves segmenting sentences into tokens, where each token represents a sequence of one or more characters [98]. Tokens in a sentence are separated by spaces [98]. In the Arabic language, tokenization presents challenges due to its rich and complex morphology [98]. In this study, we utilized the TweetTokenizer provided from the NLTK library to segment the text into tokens [99]. It was chosen because of its simple, fast tokenizing process, which focuses on Twitter data and works based on regular expressions [100]. We also chose the TweetTokenizer because of its ability to preserve the emojis and emoticons as tokens, allowing them to be dealt with appropriately, and it also handles the repeated characters by reducing them to a length of three [101], [102]. This makes it suitable for the ASTC dataset and preprocessing experiments.

7- The second stage of data preprocessing is divided into three phases based on how non-Arabic words and punctuation are handled. Since social media users often engage in code-switching practice by including non-Arabic words in their Arabic posts and tweets, and since they use punctuation mostly for aesthetic purposes without following the grammar rules, we examined the effects of handling non-Arabic words and punctuation in three experiments. The first experiment removes non-Arabic words and punctuation, the second

experiment keeps the non-Arabic words and removes the punctuation, and the third experiment keeps the non-Arabic words and the punctuation. Then, eight preprocessing conditions were tested in each experiment, denoted by R1-R8, as shown in Figure 3.1.1. These conditions are based on handling emojis, the type of the stemmer, and the embedding transformer, which are described in points 8, 9, and 10.

8- Handling Emojis: Emojis are an almost ever-present component of social media posts. They can convey strong emotions that improve the model's performance. Emojis enrich the posts with sentiment and are related to the words in the post [103]. This highlights the importance of properly handling emojis. So, we used two approaches to examine the effect of emoji handling: the first approach eliminated emojis from tweets, and the second approach translated emojis into their emotional and real meaning using the novel Emoji Meaning dataset, which depends on their usage on social media platforms by replacing each emoji with its textual meaning.

9- Stemming: is a common morphological analysis that aims to reduce inflectional forms to achieve a common base form for a word [104]. This process is important because it reduces vocabulary size and dimensionality, thus determining the weights and importance of words in a text more accurately [104], [105]. Stemming algorithms for Arabic are divided into two main types [106], [107]. Light stemmers, which remove the suffixes and prefixes from Arabic words without requiring extracting the roots, and heavy stemmers, which find the root of a word and implicitly perform light stemming [107]. Two widely used Arabic stemmers are the Information Science Research Institute (ISRI) and Snowball stemmers. The ISRI stemmer, a type of heavy stemmer, is a root-extraction stemmer designed for the Arabic language that aims to extract roots without requiring a root dictionary [108]. Conversely, the Snowball stemmer was designed to overcome the limitations of the Porter stemmer and the lack of stemmers that support multiple languages other than English [109]. The Snowball stemmer is designed to remove common suffixes without focusing on root extraction, categorizing it as a light stemmer [109]. We investigated the effect of using both types of stemmers in Arabic sentiment analysis by applying the ISRI and Snowball stemmers.

10- Embedding: The embedding technique enables the representation of words and sentences in a numerical format [110]. This is done by transforming each word into a numerical

vector representation that captures the syntactic and semantic relationships based on contextual usage within the dataset [110]. Transformation methods are divided into two main types based on the training methods: trainable models adjust the embedding vector randomly and are trained alongside the ML model for a specific task. Pertained models use word embeddings that were pre-trained on large, general datasets as input for models with other tasks. We evaluated the effect of two transformation models on model performance: The Keras embedding and the AraVec 3.0 embedding. The Keras embedding is a type of trainable embedding. It randomly adjusts the embedding vector for each word to small weights, then updates the embedding vectors during the back-propagation to minimize the loss function [111]. While AraVec 3.0 [112], on the other hand, is a free, open-source project that provides a powerful pre-trained model for the Arabic NLP research world. AraVec 3.0 is built on top of two Arabic content domains that have been utilized to train the latest version of AraVec 3.0, namely tweets and Arabic Wikipedia articles. This version provides 16 different word embedding models using CBOW and skip-gram structures, as well as two types of models: unigrams and n-grams. The most commonly used n-gram AraVec models were trained using more than 1,169 billion tokens. However, the AraVec authors did not provide details about the testing datasets and corpora [113]. In this thesis, the embedding was generated using an n-gram model with a vector size of 100.

3.5. Proposed Model

This thesis presents a combined deep learning architecture of CNN-LSTM models that was developed specifically for classifying Arabic sentiments. The proposed CNN-LSTM architecture consists of five layers, as shown in Figure 3.5.1: an embedding layer, a CNN layer, a max pooling layer, an LSTM layer, and an output layer. The CNN-LSTM model is a hybrid model that combines the advantages of both CNNs and RNNs (specifically, LSTM). Due to the effectiveness of the CNNs in extracting local features[114], [115] and the ability of LSTMs to handle long sequential data [116], this combination leverages their strengths, resulting in an effective model that can capture both local and global dependencies in text data. This makes it well-suited for handling the complexities of Arabic sentiment analysis.

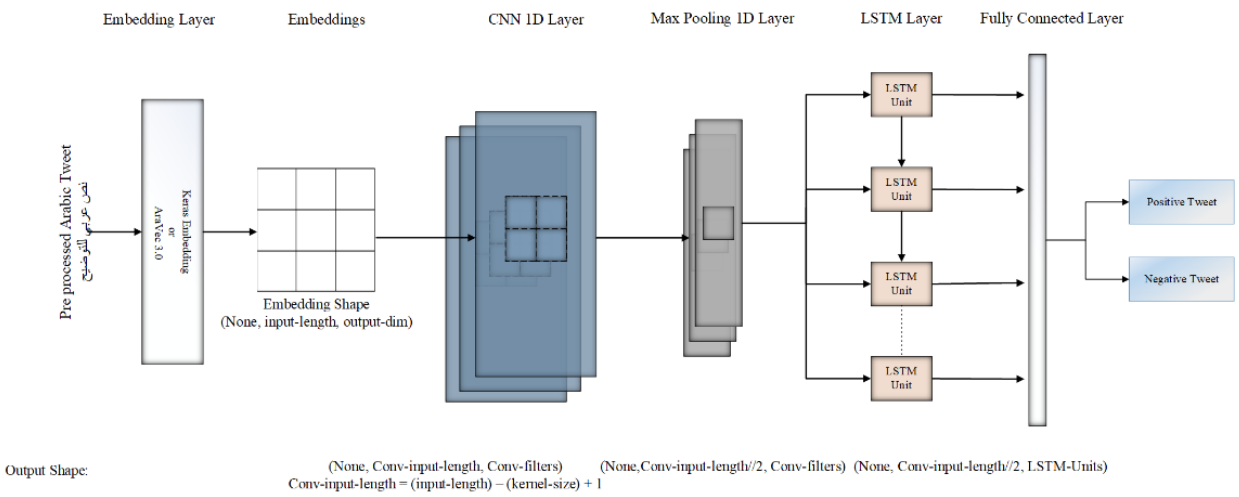

Figure 3.5.1 The proposed CNN-LSTM model architecture for Arabic sentiment analysis [117].

The first layer of the CNN-LSTM model is an embedding layer that receives a preprocessed text as an input sequence and transforms each word (token) into a dense numerical vector representation [118] to generate a matrix of dense word vectors. This allows the CNN-LSTM model to effectively process and understand the text. Embedding generation is performed in the embedding layer based on three main parameters: input-dim, which determines the vocabulary size of the dataset, output-dim, which determines the vector space to which words will be transformed, and input-length, which presents the input sequence length [119]. These parameters determine the shape of the embedding layer output, which is represented as (batch-size, input-length, output-dim). The batch-size parameter value of "None" indicates that a dynamic batch size is used, which is common in the Keras implementation [22]. The input-length parameter value varies based on the preprocessing conditions affecting the sequence length of the input data. The output dimension parameter is treated as a tunable hyperparameter with a value ranging from 100 to 400, with a 50-unit step for Keras embedding, and it has a static value of 100 for the AraVec since it is a pre-trained model with a static vector size of 100. Thus, the embedding layer generates different output shapes across different experiments and runs, depending on the input sequence length, which is affected by how non-Arabic words, punctuation, and emojis are treated (kept or removed), and the output dimension, which is determined during the hyperparameter tuning phase. Also, Table 3.5.1 summarizes the final shape of the generated embeddings for all experiments and runs. This thesis explores two types of text transformers to study their effect on the model performance in learning semantic relationships between words, which are the AraVec 3.0 and the Keras embedding.

Following the embedding layer is a 1D convolutional layer that extracts the local features [120] from the input embedding and feeds them to the max-pooling layer. Feature extraction is performed by convolving a convolutional filter (kernel) with the input matrix [119]. The kernel slides across the matrix to extract the feature map of important input data patterns [119]. So, the output of the Conv1D layer is shaped (None, Conv-input-length, Conv-filters), where the value of "None" indicates a dynamic batch size, the values of Conv-input-length that depend on the input-length and the kernel size, and the tunable hyperparameters of the convolutional filters are summarized in Table 3.5.1 for all experiments and runs. The number of filters in the Conv1D layer ranges from 100 to 400 in increments of 100. Suppose that Xi represents a K-dimensional vector for a single word in a tweet of n words [121]. This is denoted as a set of concatenated words using the "+" operator in Equation 1 [121]. $Xi{:}i{+}j$ represents the basic element of the input feature matrix that extends from the ith to $(i + j)$ words of the sentence vector [121]. When a convolutional filter with a window size of h is convolved with a window of words $Xi{:}i{+}h{-}1$, it produces a feature Cif for the same window of words, as proposed in Equation 2 [121]. The window size, also called the kernel size, was examined in this thesis with the values of 2, 3, and 6 to explore the effect of different n-gram features on model performance.

$$X_{1:n} = X_1 + X_2 + X_3 + \ldots\ldots\ldots + X_n \quad (1)$$

$$C_i = f(W.X_{i:i+h-1} + b) \quad (2)$$

The convolutional filter is represented by W, the bias, which is a real number represented by b, and the activation function by f [121]. The ReLU activation function is applied to the output of each filter in the CNN layer to help learn complex patterns. Then, convolving the filter through all the word windows will produce a feature map for a single convolutional filter based on Equation 3. When the convolutional layer has m filters, it will generate $(n{-}h{+}1)$ features [121]. Then, to prevent overfitting, the activity L2 regularization of 0.01 is used.

$$C = C_1, C_2, C_3, C_{n-h+1} \quad (3)$$

The max pooling applies the max function with a pool size of 2 to each CNN filter output to select the maximum feature value at each filter window position while iterating across the matrix [122]. This results in down-sampling the size of each feature map, reducing the output complexity, and

retaining the important features [122], [120] to feed to the LSTM layer. The final output shape, generated by the max pooling layer, is (None, Conv-input-length /2, Conv-filters). Table 3.5.1 shows the output shapes for the max pooling layer for all experiments and runs.

The LSTM layer is then used to capture long-term dependencies and address the limitations of RNNs by using a memory cell over time to understand the context, indicating emotions in text. In this thesis, an LSTM layer with a tunable dropout value was employed. The Keras tuner identified the optimal dropout rate, which varied between 0.2 and 0.5 with a step size of 0.1, to prevent overfitting. Also, the number of LSTM units is a tunable parameter used with a search space ranging from 30 to 300, with a step size of 10. This tuning range provides a trade-off between computational efficiency and performance of the model. Then, to preserve the temporal sequence for the next Flatten layer, the return_sequences parameter is set to a “True” value, which helps understand the semantic meaning, as it depends on word order. Next, the Flatten layer shapes features larger than the threshold to be suitable for input into the fully connected dense layer. The LSTM and Flatten layers' output shape is presented in Table 3.5.1.

The final layer, also called the fully connected layer [25], is a dense layer with a single unit that generates a (None, 1) output shape and a sigmoid activation function, as the task is a binary classification of positive or negative sentiment. The Adam optimizer was then used to improve the training process of the proposed model, with a learning rate that was sampled logarithmically, ranging between 1e-5 and 1e-3, to ensure optimal training convergence.

Table 3.5.1 The output shape of each layer for the 1D CNN-LSTM model.

Exp Num	Run Num	Embedding shape	Convolutional Layer	Max pooling	LSTM layer	Flatten
Exp 1	R1	(None, 1189, 150)	(None, 1187, 400)	(None, 593, 400)	(None, 593, 250)	(None, 148250)
	R2	(None, 1956, 300)	(None, 1954, 100)	(None, 977, 100)	(None, 977, 80)	(None, 78160)
	R3	(None, 1955, 150)	(None, 1953, 400)	(None, 976, 400)	(None, 976, 170)	(None, 165920)
	R4	(None, 1956, 100)	(None, 1955, 200)	(None, 977, 200)	(None, 977, 130)	(None, 127010)
	R5	(None, 968, 200)	(None, 967, 300)	(None, 483, 300)	(None, 483, 100)	(None, 48300)
	R6	(None, 1955, 100)	(None, 1953, 400)	(None, 976, 400)	(None, 976, 200)	(None, 195200)
	R7	(None, 968, 100)	(None, 963, 300)	(None, 481, 300)	(None, 481, 190)	(None, 91390)
	R8	(None, 969, 100)	(None, 964, 100)	(None, 482, 100)	(None, 482, 120)	(None, 57840)

Exp 2	R1	(None, 1328, 400)	(None, 1327, 200)	(None, 663, 200)	(None, 663, 300)	(None, 198900)
	R2	(None, 2129, 150)	(None, 2128, 400)	(None, 1064, 400)	(None,1064,270)	(None, 287280)
	R3	(None, 2128, 350)	(None, 2126, 400)	(None, 1063, 400)	(None, 1063, 230)	(None, 244490)
	R4	(None, 2129, 100)	(None, 2128, 100)	(None, 1064, 100)	(None, 1064, 110)	(None, 117040)
	R5	(None, 1141, 150)	(None, 1139, 400)	(None, 569, 400)	(None, 569, 160)	(None, 91040)
	R6	(None, 2128, 100)	(None, 2127, 200)	(None, 1063, 200)	(None, 1063, 150)	(None, 159450)
	R7	(None, 1141, 100)	(None, 1140, 400)	(None, 570, 400)	(None, 570, 190)	(None, 108300)
	R8	(None, 1142, 100)	(None, 1141, 200)	(None, 570, 200)	(None, 570, 280)	(None, 159600)
Exp 3	R1	(None, 1167, 300)	(None, 1166, 100)	(None, 583, 100)	(None, 583, 220)	(None, 128260)
	R2	(None, 2048, 350)	(None, 2046, 400)	(None, 1023, 400)	(None, 1023, 70)	(None, 71610)
	R3	(None, 2128, 400)	(None, 2127, 300)	(None, 1063, 300)	(None, 1063, 70)	(None, 74410)
	R4	(None, 2221, 100)	(None, 2219, 400)	(None, 1109, 400)	(None, 1109, 130)	(None, 144170)
	R5	(None, 1141, 400)	(None, 1139, 100)	(None, 569, 100)	(None, 569, 90)	(None, 51210)
	R6	(None, 2128, 100)	(None, 2126, 200)	(None, 1063, 200)	(None, 1063, 230)	(None, 244490)
	R7	(None, 1141, 100)	(None, 1136, 100)	(None, 568, 100)	(None, 568, 80)	(None, 45440)
	R8	(None, 1167, 100)	(None, 1166, 200)	(None, 583, 200)	(None, 583, 230)	(None, 134090)

Additionally, each layer of the CNN-LSTM model has a certain number of trainable parameters that update during the training phase. Table 3.5.2 shows the number of trainable parameters for each layer of the CNN-LSTM model across all experiments and runs. This provides a clear picture of the model's complexity and its ability to learn from the data. The number of trainable parameters can be automatically calculated for each layer using the Keras model. summary () function.

Table 3.5.2. The number of trainable parameters for each CNN-LSTM layer.

Exp Num	**Run Num**	**Embedding Param**	**Conv Layer Param**	**Max pooling Param**	**LSTM layer Param**	**Flatten Param**	**Dense layer Param**
Exp 1	R1	10,774,200	180,400	0	651,000	0	148,251
	R2	5,438,400	90,100	0	57,920	0	78,161
	R3	4,934,700	180,400	0	388,280	0	165,921
	R4	147,671,600	40,200	0	172,120	0	127,011
	R5	6,217,000	120,300	0	160,400	0	48,301
	R6	147,671,600	120,400	0	480,800	0	195,201

	R7	147,671,600	180,300	0	373,160	0	91,391
	R8	147,671,600	60,100	0	106,080	0	57,841
Exp 2	R1	30,356,400	160,200	0	601,200	0	198,901
	R2	2,938,350	120,400	0	724,680	0	287,281
	R3	12,019,700	420,400	0	580,520	0	244,491
	R4	147,671,600	20,100	0	92,840	0	117,041
	R5	4,897,950	180,400	0	359,040	0	91,041
	R6	147,671,600	40,200	0	210,600	0	159,451
	R7	147,671,600	80,400	0	449,160	0	108,301
	R8	147,671,600	40,200	0	538,720	0	159,601
Exp 3	R1	5,450,400	60,100	0	282,480	0	128,261
	R2	6,902,000	420,400	0	131,880	0	71,611
	R3	13,765,200	240,300	0	103,880	0	74,411
	R4	147,671,600	120,400	0	276,120	0	144,171
	R5	13,086,800	120,100	0	68,760	0	51,211
	R6	147,671,600	60,200	0	396,520	0	244,491
	R7	147,671,600	60,100	0	57,920	0	45,441
	R8	147,671,600	40,200	0	396,520	0	134,091

3.6. Tuning Parameters

Hyperparameters are a set of parameters that determine the structure and behavior of a model [123]. The systematic search process used to identify the best hyperparameter combinations to improve model robustness and performance is called Hyperparameter Optimization (HPO) [123]. The HPO plays a vital role in preventing overfitting, speeding up the computational process, and minimizing the costs, especially when using automatic approaches and frameworks [123]. Thus, the Keras Tuner [124] was used in the final phase to tune the model [124]. The Keras Tuner provides an easy-to-use, scalable optimization framework for deep learning models [124]. It solves the critical hyperparameter search issues by providing several built-in search algorithms (Bayesian Optimization, Random Search, and Hyperband) and by configuring the search space using a define-by-run syntax to define parameter value ranges to find the optimal hyperparameter values for improving the model's validation evaluation [124], [123]. In this thesis, the Keras Tuner was employed with Bayesian optimization as the search algorithm to find the optimal hyperparameters, focusing on promising regions and reducing the number of required trials. Hyperparameter selection is based on its effect on model accuracy during the five-fold cross-validation process.

Thus, the best model was selected for its consistency across different folds and for its validation performance.

During the 5-fold cross-validation, the Keras tuner was used to validate the model and optimize the hyperparameters using five different validation folds. The training data was divided into five folds: four for training and one for validation. This distribution balances bias and variance in error estimation, reduces computational effort (since the hybrid model is computationally expensive), and preserves robust performance estimation. This also allowed the model to explore different hyperparameter combinations during each fold and select the ones that produced the best overall performance. Additionally, during the training and validation processes, all previously defined dynamic batch sizes were set to a value of 50 after manually exploring different values, as the model achieved the best results with this value. Finally, the model with the best validation performance across all folds was evaluated using the same number of epochs (10) on test data that had never been seen before.

3.7. Model Evaluation

After the modeling stage, which involved training and validating the hybrid model, several performance metrics were used to check and evaluate the model's performance in the Arabic sentiment prediction task. These metrics evaluated the consistency of the CNN-LSTM based on the match between the predicted and actual values of the test data after training the hybrid model. Although accuracy is the most widely used performance metric, it may not fully capture the evaluation [125]. Therefore, several other performance metrics were used, including precision, recall, the F1-score, confusion metrics, and the Area Under the Curve (AUC), to ensure a comprehensive evaluation.

Accuracy is a widely used performance metric representing an overall measure of how often the model correctly classifies sentiments into their respective classes. Accuracy is the ratio of correctly predicted observations to the total number of instances. It is calculated using Equation 4, which is represented mathematically as the sum of the True Positive (TP) and True Negative (TN) divided by the sum of True Positive, True Negative, False Positive (FP), and False Negative (FN).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Accuracy provides an appropriate performance metric when the data are balanced across classes. However, in the case of imbalanced datasets, accuracy can be misleading because the model may predict the majority class with high accuracy while failing to predict the minority class [126]. This limitation underscores the importance of considering additional performance metrics, such as precision, recall, the F1-score, confusion metrics, and the AUC, to gain a more nuanced understanding of model performance [127].

Precision is an evaluation metric that indicates model performance by finding the ratio of instances correctly predicted as positives to the total predicted positives. It is represented mathematically as TP divided by the sum of TP and FP, which represents all observations predicted as positives, as shown in Equation 5. Precision is an important metric, especially when false positives have a high cost. High precision indicates that the model correctly predicts most of the positive sentiments. However, focusing on increasing precision may lead to missed positive instances (false negatives), and relying on precision alone can lead to an incomplete model evaluation [128]. Thus, it is important to combine precision with recall to provide a better model evaluation [128].

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall, also known as sensitivity, is a popular metric that assesses the consistency of a model's performance by determining the ratio of instances correctly predicted as positive instances to all actual positives. This ratio is represented mathematically by dividing the TP by the sum of TP and FN instances, as shown in Equation 6. A high recall score means that the model successfully predicts most actual positive observations, which is useful in cases when missing a positive observation has a high cost. Furthermore, to evaluate models more effectively, the recall metric is best interpreted alongside the precision metric.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

F1-score is an important metric for verifying test accuracy, as it calculates the harmonic mean of precision and recall [129]. Thus, the importance of the F1-score rises in cases where the data distribution is imbalanced, as it balances false positives and false negatives. The F1-score calculation is shown in Equation 7.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (7)$$

A Confusion Matrix is a performance measurement tool that displays the classification results as the number of true positives, true negatives, false positives, and false negatives. For binary classification tasks, the predicted instance has these four possible options, giving the confusion matrix the ability to compute several evaluation metrics [130]. These metrics include accuracy, precision, recall, sensitivity, specificity, and F1-score [131]. A confusion matrix is represented by a square matrix of size N × N, where N is the number of output classes [131]. As shown in Table 3.6.1, the confusion matrix consists of four parameters that reflect the model's ability to correctly predict the classes. A high count of TP and TN and a lower count of FP and FN improve model performance, and vice versa. These parameters are summarized as follows:

- True Positive: actual positive instances are predicted as positive.
- False Positive: actual instances that are negative are predicted by the model as positive.
- True Negative: actual instances that are negative are predicted by the model as negative.
- False Negative: actual instances are positive, but the model predicts them as negative.

Table 3.6.1 Confusion Matrix

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC)

The AUC-ROC is a measure of a classifier's ability to distinguish between the two classes [132]. The ROC plots and compares two operating characteristics: the True Positive Rate (TPR) and the False Positive Rate (FPR) for each threshold, and then the area under the ROC curve is computed [132], [131], [133]. A high AUC indicates an accurate classifier, while a low AUC indicates an inaccurate classifier [131]. The AUC ranges from 0 to 1, where classifiers with an AUC score close to 1 are accurate, while those with an AUC score of 0.5 or lower are inaccurate and unable to distinguish between classes. Figure 3.6.1 illustrates how the AUC varies using three curved lines

drawn between the (0, 0) and (1, 1) points, where the y-axis presents the TPR and the x-axis presents the FPR.

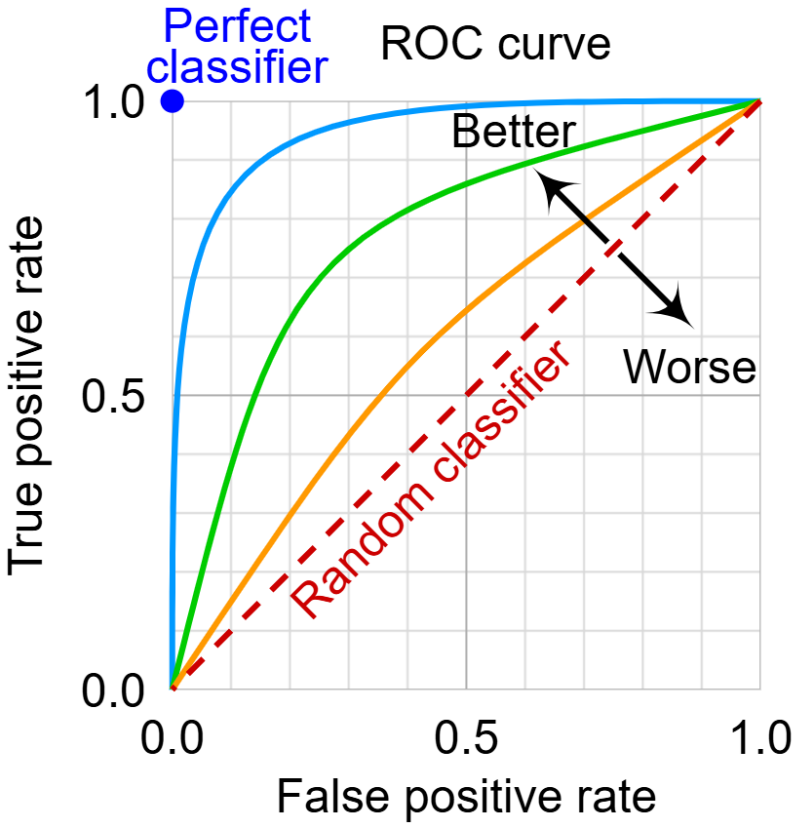

Figure 3.6.1 ROC and AUC illustration [134].

Chapter 4 Experiments and Results

4.1. Introduction

This chapter presents the results of the combined deep learning model, which demonstrate the model's performance and explain the effect of each data preprocessing step on the model's performance, including the effect of keeping and removing the non-Arabic words and punctuation, which were distributed across three experiments and examined over eight combinations of preprocessing conditions. These preprocessing conditions show the importance of encoding the emojis with their real and emotional meanings and finding an appropriate combination of stemmer and embedding transformer for each group of preprocessing steps. In each experimental condition, the performance metrics, including accuracy, precision, recall, and AUC, are applied to determine the best model. These conditions were categorized into eight groups named from R1 to R8 to improve the ASA as described in Table 4.1.1.

Table 4.1.1. Groups of data preprocessing conditions.

Experiment Conditions	Stemmer Name	Emoji Handling	Embedding Transformer
R1	ISRI stemmer	emoji removal	Keras embedding
R2	ISRI stemmer	emoji encoding	Keras embedding
R3	Snowball	emoji encoding	Keras embedding
R4	ISRI stemmer	emoji encoding	AraVec 3.0
R5	Snowball	emoji removal	Keras embedding
R6	Snowball	emoji encoding	AraVec 3.0
R7	Snowball	emoji removal	AraVec 3.0
R8	ISRI	emoji removal	AraVec 3.0

The training process of the hybrid model was controlled by the set of hyperparameters initiated and updated by the Keras tuner using the Bayesian optimization technique. Then, the model performance was evaluated using the validation data. These operations are repeated based on a predefined number of iterations to select the best model with the best results. In addition, 5-fold

cross-validation was used during model training to ensure that the model was consistent across different subsets of the data and to avoid overfitting and underfitting. As a result, the best hyperparameters for each experiment are summarized in Table 4.1.2.

Table 4.1.2 The best hyperparameter values are determined by the Keras tuner.

Exp Num	Run	Output Dim	Convolutional Filters	Convolutional Kernel Size	LSTM Units	LSTM Dropout	Learning Rate
Exp 1	R1	150	400	3	250	0.3	0.00049751
	R2	300	100	3	80	0.3	0.00017099
	R3	150	400	3	170	0.2	0.00022555
	R4	100	200	2	130	0.4	0. 00023515
	R5	200	300	2	100	0.4	0. 00014636
	R6	100	400	3	200	0.3	0.00001149
	R7	100	300	6	190	0.3	0.00002469
	R8	100	100	6	120	0.4	0.00002816
Exp 2	R1	400	200	2	30	0.2	0. 00016625
	R2	150	400	2	270	0.2	0.00007771
	R3	350	400	3	230	0.4	0.00053589
	R4	100	100	2	110	0.2	0.00007517
	R5	150	400	3	160	0.3	0.00019196
	R6	100	200	2	150	0.3	0.00043968
	R7	100	400	2	190	0.2	0.00007383
	R8	100	200	2	280	0.2	0.000144206
Exp 3	R1	300	100	2	220	0.2	0. 00028067
	R2	350	400	3	70	0.4	0. 00027495
	R3	400	300	2	70	0.2	0.00022570
	R4	100	400	3	130	0.3	0.00003997
	R5	400	100	3	90	0.2	0.00009749
	R6	100	200	3	230	0.2	0.00001491
	R7	100	100	6	80	0.4	0.00007121
	R8	100	200	2	230	0.2	0.00011200

The hyperparameter values differ between experiments to achieve the best results, with no repeating pattern of hyperparameter values. Therefore, it is not possible to generalize a set of these values to be applied to all experiments and achieve the same results. Instead, the Keras tuner was used to determine the optimal set of values for each parameter in each experiment, as appropriate. The results of all three experiments are summarized in Tables 4.2.1, 4.3.1, and 4.4.1. All experiments were performed on the Google Colab-L4 platform using Python version 3.10.12. The deep learning models were implemented using Keras version 3.4.1, running on TensorFlow version 2.17.0.

The rest of the chapter is organized as follows: Sections 4.2, 4.3, and 4.4 present the results of experiments one, two, and three for all preprocessing conditions. They display the numerical

scores of precision, recall, accuracy, and F1-score, as well as providing visual representations of these results to simplify understanding of model performance. Additionally, a confusion matrix was presented for each experiment to give a deeper understanding of the model's ability to predict positive and negative sentiments. Section 4.5 presents a detailed discussion of the model results, comparing them with other approaches applied to the same dataset.

4.2. Experiment One

In this experiment, the data underwent aggressive preprocessing steps, where the effect of removing non-Arabic words and punctuation was tested over eight conditions, resulting in eight experimental runs. These experiments show the power of embedding transformers and stemmers in exploring the insight from the Arabic text only, which will show an important distinction between trainable transformers trained only on the current dataset and pre-trained transformers and the different stemmers. In addition, these experiments show the ability of these technologies to deal with punctuation and their use in providing better vector representation. Table 4.2.1 presents all the results for these experimental conditions:

Table 4.2.1 Experiment One results.

Run	Stemmer	Emoji	Embedding	Precision	Recall	F1-Score	Accuracy
R1	ISRI	Remove Emoji	Keras Embedding	72%	70%	70%	70.15%
R2	ISRI	Encoding to Arabic	Keras Embedding	90%	90%	90%	90.23%
R3	Snowball	Encoding to Arabic	Keras embedding	91%	91%	91%	91.69%
R4	ISRI	Encoding to Arabic	AraVec 3.0	87%	87%	87%	87.32%
R5	Snowball	Remove Emoji	Keras Embedding	70%	70%	70%	69.92%
R6	Snowball	Encoding to Arabic	AraVec 3.0	76%	76%	76%	76.09%
R7	Snowball	Remove Emoji	AraVec 3.0	54%	54%	53%	53.87%
R8	ISRI	Remove Emoji	AraVec 3.0	54%	53%	53%	53.55%

The results of the first experiment in Table 4.2.1 proposed a set of facts, including the rich sentiments contained in the emojis that were displayed by removing the emojis in R1, R5, R7, and R8, which had a negative impact on the model performance, achieving the lowest accuracies of 70.15%, 69.92%, 53.87%, and 53.87%, respectively. While emojis encoding to their textual meaning, which includes the real and emotional meaning, results in improving the model classification performance in R2, R3, R4, and R6, by achieving accuracies of 90.23%, 91.69%, 87.32%, and 76.09%, respectively. Also, the results in R2 and R3 prove that the Keras transformer

is better than the AraVec transformer for both stemmers. At the same time, the AraVec embedding achieves a lower bit accuracy of 87.32% in R4 when using the ISRI stemmer and a poor performance when using the Snowball stemmer. Moreover, the Keras transformer generates a better vector representation when removing the emojis in R1 and R5 for both stemmers, while in R7 and R8, the AraVec cannot generate a meaningful vector representation to support the classification task, resulting in poor accuracies of 53.87% and 53.55%, respectively. This can be explained by the fact that the Keras transformer has the advantage over the AraVec transformer of being specifically trained on the same dataset.

The data visualization technique helps in presenting data in a clear pictorial or graphical format, which makes the information easy to understand [135]. Figure 4.2.1 uses the bar plot to provide another point of view for the performance results.

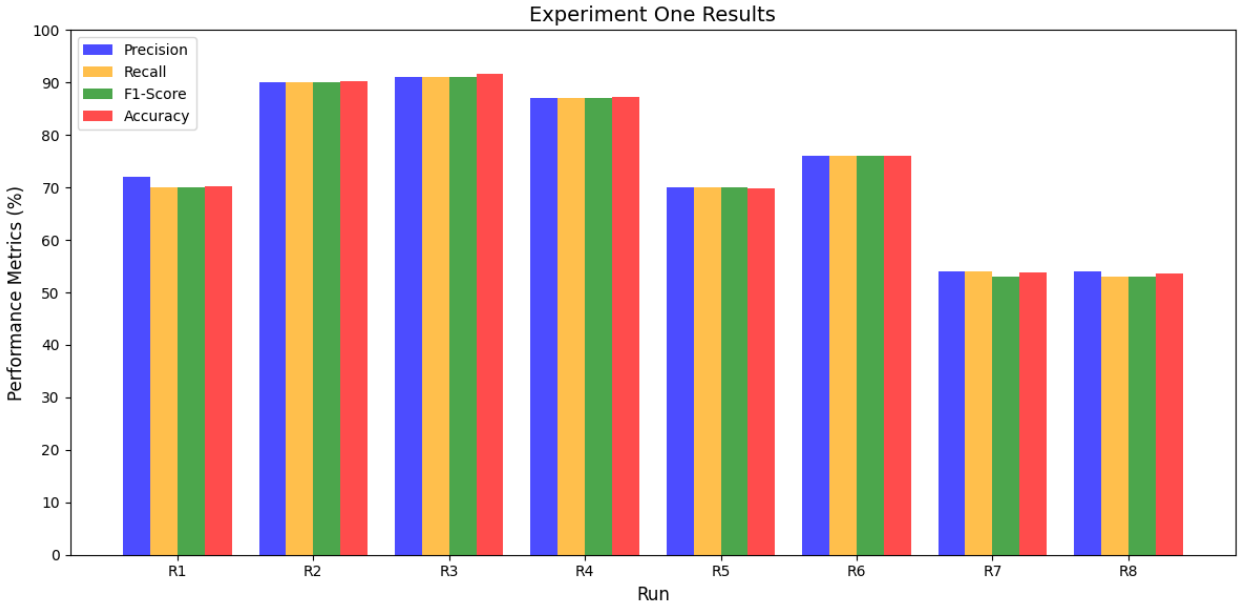

Figure 4.2.1 Visual representation of Experiment One performance results.

Figure 4.2.1 helps in presenting two facts. The first fact ensures that R3 and R2 provide the highest performance results that show the outperformance of the Keras transformer over the AraVec transformer. The second fact shows that all performance metrics for each experiment run has close results, indicating that the hybrid model performance is a well-balanced model performs consistently across all the evaluation metrics.

Also, Figure 4.2.2 shows the confusion matrix for each experiment. This helps to examine and evaluate the data preprocessing strategies and reveals several trends correlated with these conditions.

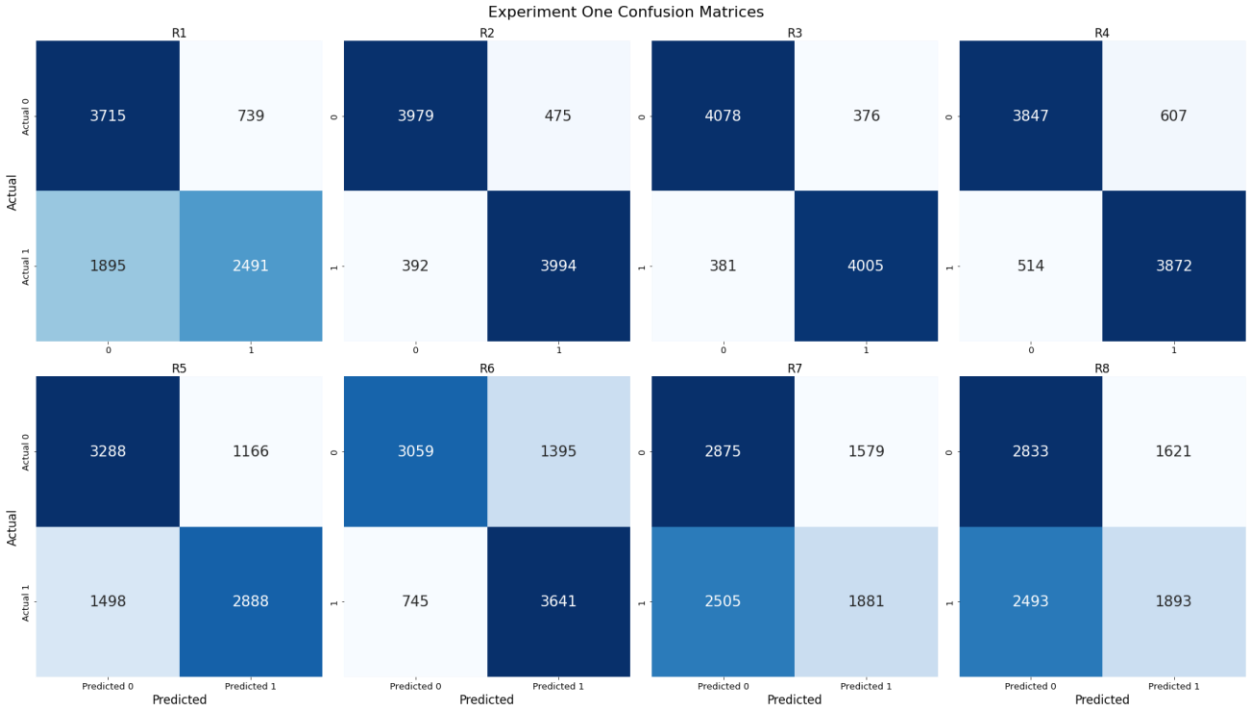

Figure 4.2.2 Experiment One confusion matrices.

For R1 and R5, the matrices show a higher number of false negatives cases (FN = 1895) for R1 and (FN = 1498) for R5 compared to the false positives cases (FP = 739) for R1 and (FP = 1166) for R5. This indicates that the hybrid model struggles to correctly classify the positive sentiments. Also, for R7 and R8, the matrices show a high number of both false negative and false positive cases (FN = 2505, FP = 1579) and (FN = 2493, FP = 1621), respectively. This indicates the struggle of the model's ability to correctly classify the positive and negative sentiments, likely due to the loss of the information embedded in the emojis for the R1, R5, R7, and R8. On the other hand, for R2, R3, R4, and R6, the model shows significant improvements as the FN and FP decreased dramatically when the emojis were encoded, reaching its lower level (FN = 381, FP = 376) for the R3 and higher level (FN = 745, FP = 1395) for the R6. This can be interpreted as the role of emojis in enriching the Arabic text with emotions and the ability of the Keras transformer in R2 and R3 over the AraVec transformer in R4 and R6.

Figure 4.2.3 shows a ROC-Curve for Experiment One R3 because it achieves the best accuracy score of all Experiment One runs. The ROC-Curve shows the same results as the confusion matrix in Figure 4.2.2 above. The curve rises dramatically to the upper left near the Y-axis, showing high true positive and true negative rates. This highlights the model's robustness and balance in distinguishing between positive and negative sentiments. These contribute to its overall superior performance, resulting in an area under the ROC-Curve of 91.44%.

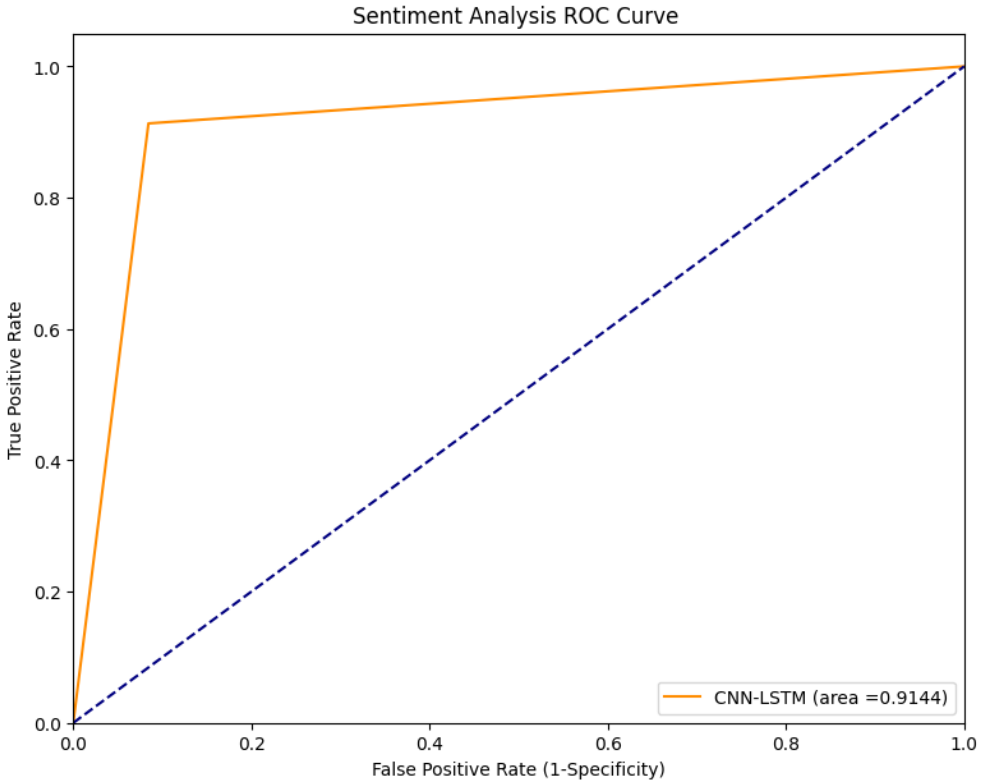

Figure 4.2.3 Experiment One R3 ROC-Curve.

4.3. Experiment Two

In this experiment, the impact of keeping the non-Arabic tokens and removing the punctuation was tested using eight parameter combinations, resulting in eight experimental runs. These experiments show the difference between the ability of embedding transformers and stemmers in extracting insight from the Arabic text with words and sentences in other languages, which will show an

important distinction between trainable transformers, which are trained only on the current dataset, and pre-trained transformers and the different stemmers. In addition, these experiments show the effect of removing the punctuation on the performance of these technologies and in providing a vector representation of the tweets. Table 4.3.1 shows all the results for these experimental conditions:

Table 4.3.1 Experiment Two results.

Run	Stemmer	Emoji	Embedding	Precision	Recall	F1-Score	Accuracy
R1	ISRI	Remove Emoji	Keras embedding	70%	70%	70%	70.11%
R2	ISRI	Encoding to Arabic	Keras embedding	90%	90%	90%	89.81%
R3	Snowball	Encoding to Arabic	Keras embedding	92%	92%	92%	91.85%
R4	ISRI	Encoding to Arabic	AraVec 3.0	78%	78%	78%	78.08%
R5	Snowball	Remove Emoji	Keras Embedding	70%	70%	70%	69.94%
R6	Snowball	Encoding to Arabic	AraVec 3.0	76%	75%	75%	75.59%
R7	Snowball	Remove Emoji	AraVec 3.0	57%	57%	56%	56.48%
R8	ISRI	Remove Emoji	AraVec 3.0	57%	56%	56%	56.07%

The second experiment results in Table 4.3.1 show the effect of using the Emoji Meaning dataset in the sentiment analysis task by removing the emojis in R1, R5, R7, and R8, which negatively affected the model performance and achieved the lowest accuracies of 70.11%, 69.94%, 56.48%, and 56.07%, respectively. At the same time, encoding the emoji based on the Emoji Meaning dataset to their textual meaning improved the model classification performance due to the richness of the tweets meaning with the emotions and real meaning inside the emojis in R2, R3, R4, and R6, achieving accuracies of 89.81%, 91.85%, 78.08%, and 75.59%, respectively. Also, when encoding the emojis, Keras embedding in R2 and R3 proved to be better than the AraVec in R4 and R6 for both stemmers. Also, it gives a better vector representation when eliminating emojis in R1 and R5, while the AraVec transformer suffers from a lack of ability to represent the emotions in R7 and R8. In addition, the keeping of the non-Arabic words in this experiment showed the superior ability of the Snowball stemmer and the Keras embedding combination in dealing with other languages over the ISRI stemmer, the AraVec embedding, and other combinations, as shown by the results in R2, R3, R4, and R6. This can be interpreted due to the fact that Keras was trained on the used dataset, which contains the Arabic tweets that may have different words or sentences written in different languages. Meanwhile, AraVec is a pre-trained transformer based on Arabic

datasets that gives Keras the outperform in dealing with these words in the testing part to achieve the best performance in the sentiment classification task.

In order to provide a clearer representation of the results, the graphical format used as a bar plot in Figure 4.3.1 allows for a deeper look at the results and makes a comparison between the performance results of the eight conditions for this experiment.

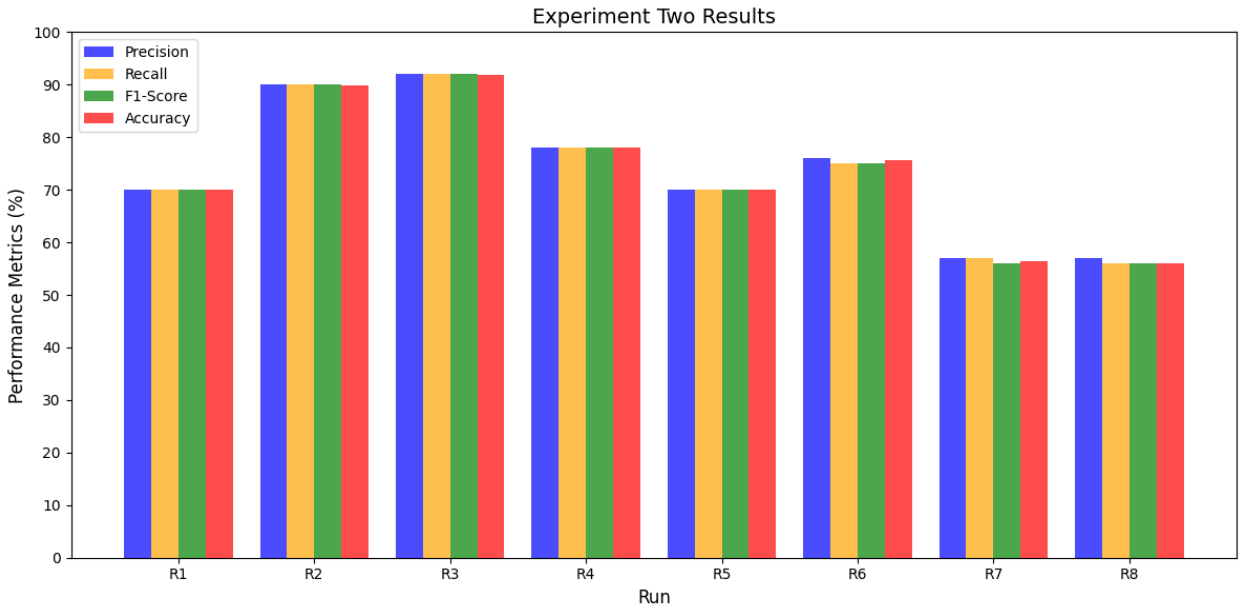

Figure 4.3.1 Visual representation of Experiment Two performance results.

The performance metrics presented by the bar chart in Figure 4.3.1 help to understand that the experiments in R3 and R2 obtained the best results when encoding the emojis, and in R1 and R5 when eliminating the emojis, which ensures that the Keras transformer outperforms the AraVec when using both the ISRI and Snowball stemmers. Also, the performance metrics presented in the bar chart show the consistency of the CNN-LSTM model for this experiment under the eight preprocessing conditions, since precision, recall, F1-score, and accuracy are very close in value.

Confusion matrices in Figure 4.3.2 display another presentation of performance measures for Experiment Two. It examines the data preprocessing strategies and presents a detailed insight into the model classification performance.

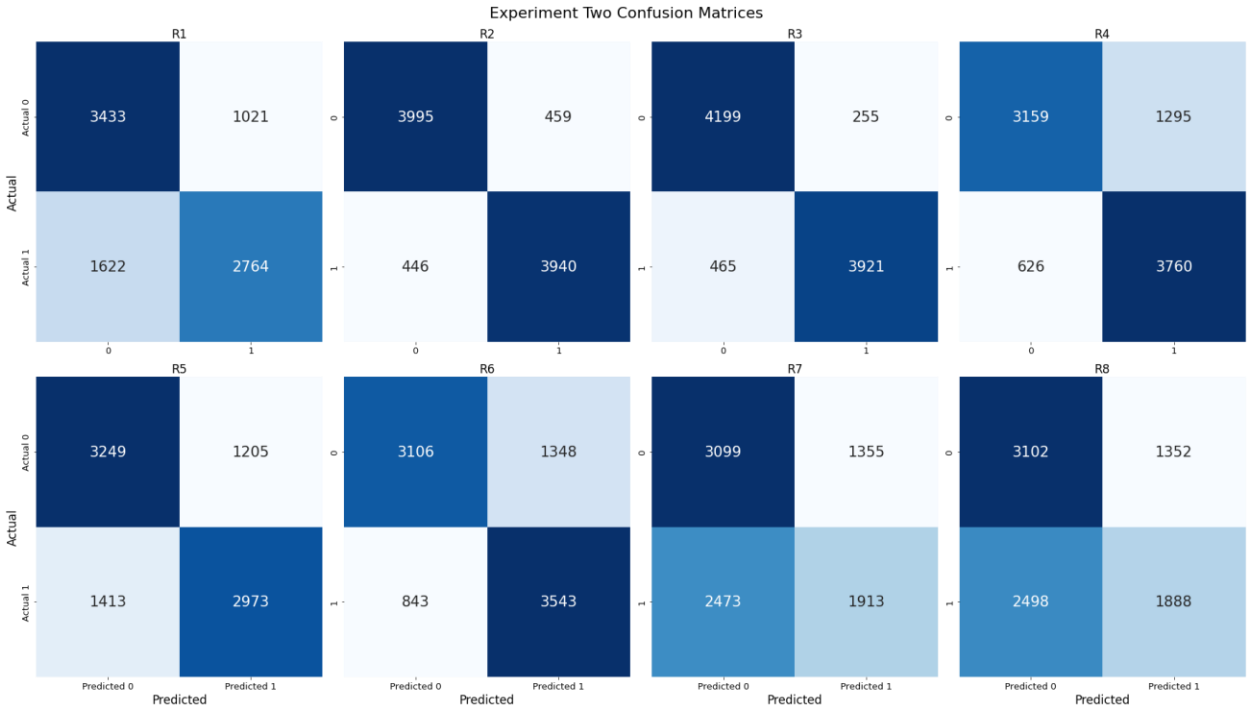

Figure 4.3.2 Experiment Two confusion matrices.

For experiments R1 and R5, the matrices show a high number of FN and FP cases (FN = 1622, FP = 1021) for R1 and (FN = 1413, FP = 1205) for R5. Also, for R7 and R8, the matrices show a higher number of both false negative and false positive cases (FN = 2473, FP = 1355) and (FN = 2498, FP = 1352), respectively, which shows the lack of ability of the AraVec transformer, and they interpret the low-performance scores obtained. Thus, the results of R1, R5, R7, and R8 indicate the struggle of the model's ability to correctly classify the positive and negative sentiments, probably due to the loss of emotions after eliminating the emojis. On the other hand, for R2 and R3 the model shows significant improvements as the FN and FP decreased dramatically, reaching their lower level (FN = 465, FP = 255) for the R3 and increased when using the ISRI stemmer instead of the Snowball one with the Keras embedding (FN = 446, FP = 459) for the R2. Results of R2, R3, R4, and R6 can be interpreted as the role of emojis in enriching the Arabic text with emotions. Also, the ability of the Keras transformer in R2 and R3 over the AraVec transformer, which has a lower ability in extracting the insights from the foreign words retained in the text in R4 and R6, with a moderate level of FN and FP scores (FN = 626, FP = 1295) and (FN = 843, FP = 1348), respectively.

Figure 4.3.3 shows the ROC-Curve for Experiment Two R3 because it obtained the best performance score of all Experiment Two runs. The ROC-Curve shows the same results as the

confusion matrix in Figure 4.3.2 above. The curve rises dramatically to the upper left near the Y-axis to show the high true positive and true negative rates, highlighting the robustness and the balance of the model in distinguishing between positive and negative sentiments. This contributes to the model's superior performance, resulting in an area under the ROC-Curve of 91.84%.

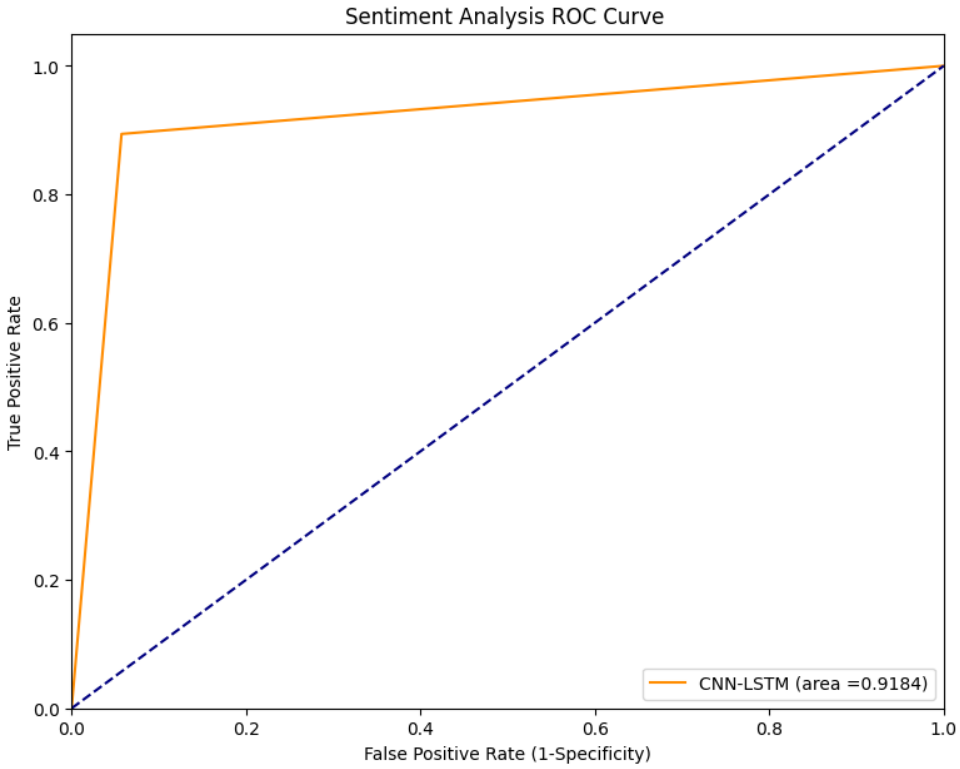

Figure 4.3.3 Experiment Two R3 ROC-Curve

4.4. Experiment Three

In this experiment, the effect of keeping the non-Arabic words and punctuation was tested over eight conditions, resulting in eight experimental runs. These experiments show the power of embedding transformers and stemmers in finding a useful vector representation from the Arabic text that has the punctuation and words written in other languages. This will show an important distinction between trainable transformers, which are trained only on the current dataset, and pre-trained transformers and the different stemmers. In addition, these experiments show the ability of these technologies to deal with punctuation and non-Arabic words, and their ability to provide a better vector representation. Table 4.4.1 shows all the results for these experimental conditions:

Table 4.4.1 Experiment Three results.

Run	Stemmer	Emoji	Embedding	Precision	Recall	F1-Score	Accuracy
R1	ISRI	Remove Emoji	Keras embedding	71%	70%	69%	69.79%
R2	ISRI	Encoding to Arabic	Keras embedding	91%	91%	91%	90.28%
R3	Snowball	Encoding to Arabic	Keras embedding	90%	90%	90%	90.43%
R4	ISRI	Encoding to Arabic	AraVec 3.0	77%	77%	77%	77.01%
R5	Snowball	Remove Emoji	Keras Embedding	70%	70%	70%	70.03%
R6	Snowball	Encoding to Arabic	AraVec 3.0	78%	77%	77%	77.46%
R7	Snowball	Remove Emoji	AraVec 3.0	54%	54%	53%	54.14%
R8	ISRI	Remove Emoji	AraVec 3.0	55%	55%	55%	55.1%

The third experiment results in Table 4.4.1 also show the importance of translating the emojis in R2, R3, R4, and R6. This provides a real improvement of the model results, achieving an accuracy of 90.28%, 90.43%, 77.01%, and 77.46%, respectively, over the results in R1, R5, R7, and R8, which are directed towards eliminating the emojis to achieve an accuracy of 69.79%, 70.03%, 54.14%, and 55.1%, respectively. These results, especially in R2 and R3 when encoding the emojis and in R1 and R5 when eliminating them, show that Keras embedding outperforms the AraVec transformer when using ISRI or Snowball stemmers. In addition, the keeping of the non-Arabic words and punctuation in this experiment shows the superior ability of the Snowball stemmer and the Keras embedding to deal with other languages over the ISRI stemmer and the AraVec embedding, as in Experiments 1 and 2. However, keeping the punctuation has no effect on improving the results; instead, the results in this experiment decreased when the punctuation was kept compared to the results in Experiments 1 and 2, except in R2 and R6. These results reflect that tweets are written with punctuation to decorate the text and that Twitter users do not follow the rules of the Arabic language. It also focuses on the importance of removing punctuation from tweets to obtain real and more accurate results for the SA. In this experiment, the best results of 90.43% were obtained in R3 when using Keras embedding, Snowball stemmer, and emoji encoding, which reflects their ability to deal with the punctuation and extract the emotions expressed in the place of the emojis.

The bar plot in Figure 4.4.1 provides a graphical format of performance results, allowing for a deeper understanding of the results and making a comparison between the performance results of the eight conditions for this experiment using the graphical techniques.

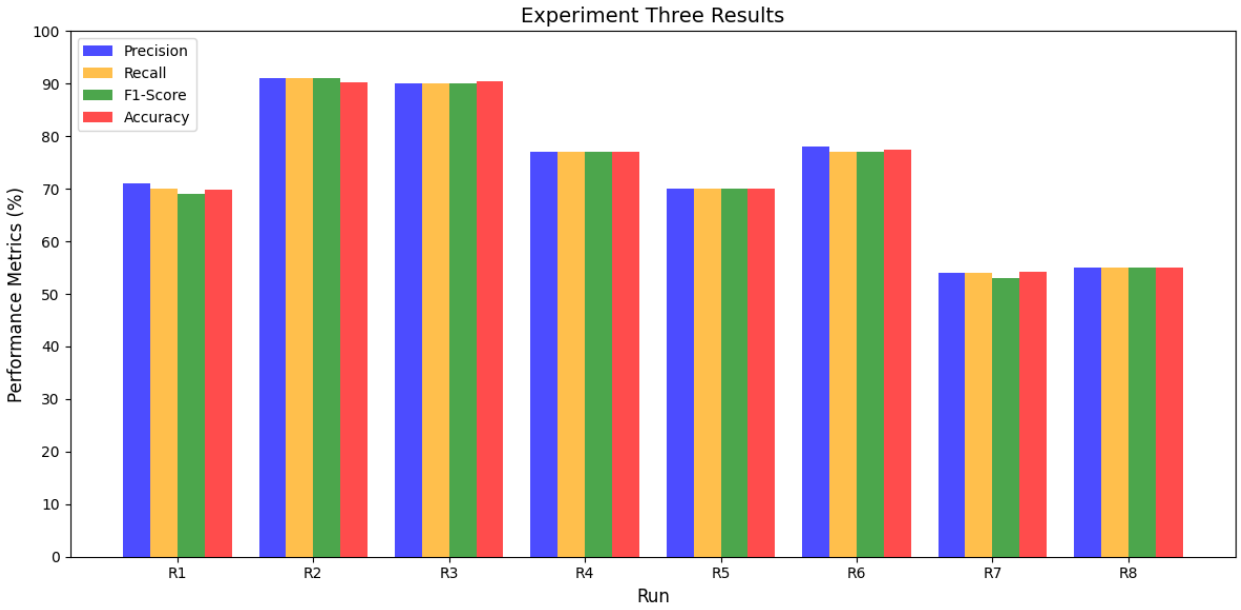

Figure 4.4.1 Visual representation of Experiment Three performance results.

Figure 4.4.1 presents the results of the experiments when the non-Arabic words and punctuations are kept. This ensures two facts about the model, in R2 and R3, the model obtained the best results when translating the emojis, and in R1 and R5, when just removing the emojis. These results show that the Keras transformer outperforms the AraVec when using both the ISRI and Snowball stemmers. Also, the performance metrics presented in the bar chart show the consistency of the CNN-LSTM model for this experiment under the eight preprocessing conditions, since precision, recall, F1-score, and accuracy are very close in value to each other, indicating the high true positive rate and true negative rate of the model.

Figure 4.4.2 shows the confusion matrices in Experiment Three, where the non-Arabic words and punctuation were retained to present detailed information about the model performance.

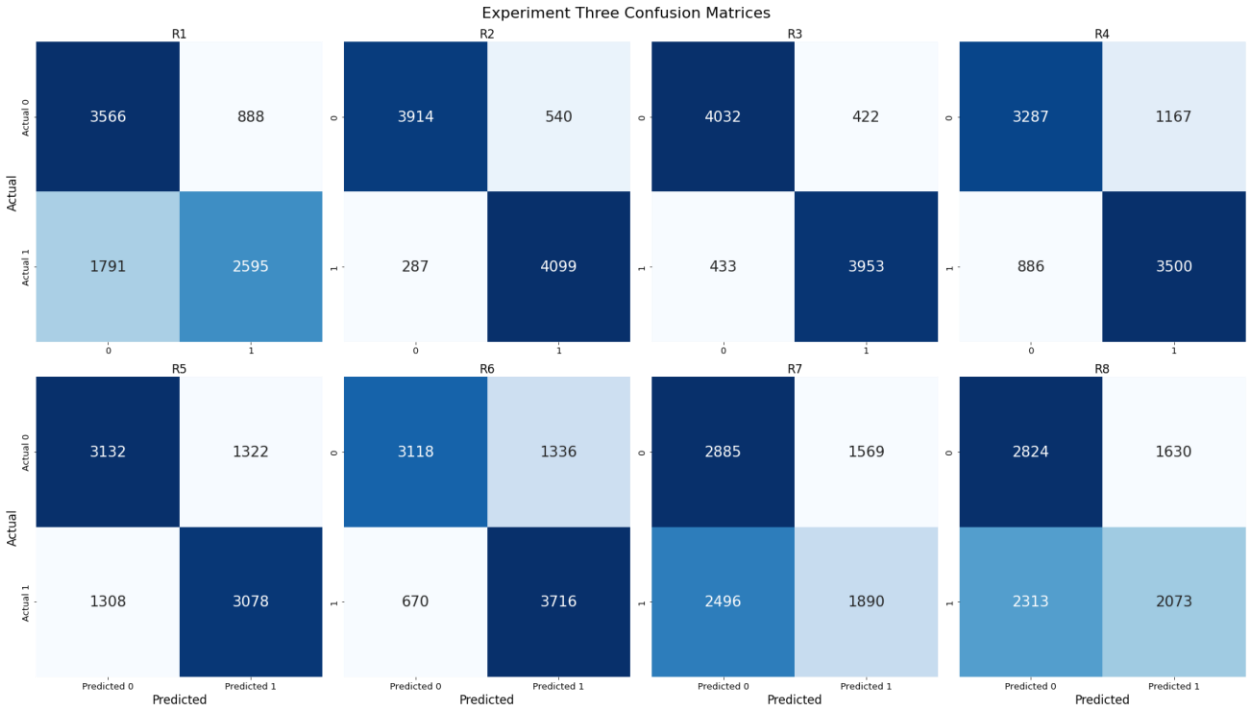

Figure 4.4.2 Experiment Three confusion matrices.

The confusion matrices in Experiment Three in Figure 4.4.2 show the role of the retention of the non-Arabic words and punctuation on the performance results when the emojis are eliminated and retained. For R1 and R5, the matrices show a high number of FN and FP cases (FN = 1791, FP = 888) for R1 and (FN = 1308, FP = 1322) for R5, while for R7 and R8 it increased dramatically to interpret the side effect of using the AraVec transformer even when retaining the punctuation and non-Arabic words, since the FN and FP scores are (FN = 2496, FP = 1569) and (FN = 2313, FP = 1630), respectively. This indicates that there is a large amount of emotion in the emojis, and when they are eliminated, the model has difficulty classifying positive and negative tweets. While the confusion matrices show a balanced result for R2 and R3, the FN and FP have the lowest level across all runs in this experiment in R3 (FN = 433, FP = 422) followed by R2 with (FN = 287, FP = 540) to show that the Keras embedding with both the ISRI and Snowball stemmer performs better than the AraVec embeddings in R4 and R6, especially when using the emoji encoding technique to achieve a score of (FN = 886, FP = 1167) for R4 and (FN = 670, FP = 1336) for R6. The ROC-Curve in Figure 4.4.3 for Experiment Three R3, since it obtained the best performance score of all Experiment Three runs. The ROC-Curve reflects the same results as the confusion matrix in Figure 4.4.2 above. The curve rises dramatically to the upper left near the Y-axis,

showing the high true positive and true negative rates. This highlights the model's robustness and balance in distinguishing between positive and negative sentiments, contributing to its overall superior performance, resulting in an area under the ROC-Curve of 90.33%.

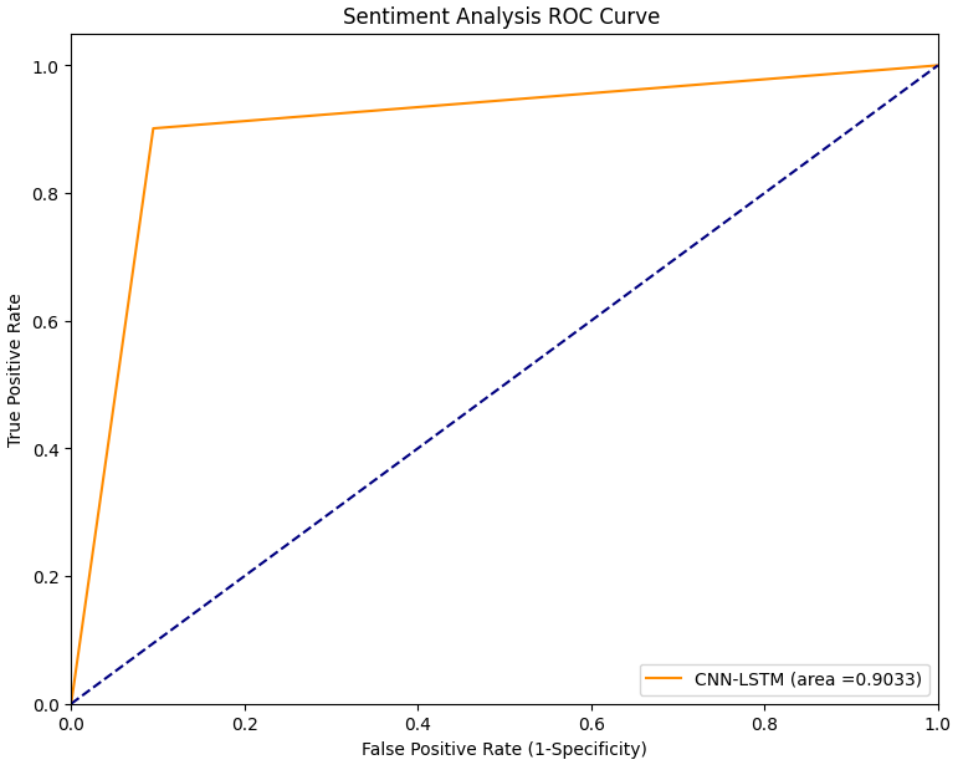

Figure 4.4.3 Experiment Three R3 ROC-Curve

4.5. Discussions

This thesis introduces a combined CNN-LSTM approach for analyzing and classifying Arabic tweets. The combined deep learning model was also utilized to investigate the role of preprocessing strategies in enhancing Arabic sentiment analysis by comparing the performance scores of different preprocessing groups to identify the most suitable set of preprocessing steps for the ASTC tweets dataset. Additionally, the impact of translating emojis into their meanings was examined to determine their significance in data preparation beyond traditional emoji handling methods.

The first experiment involved removing all non-Arabic words and punctuation. Then, several data preprocessing techniques for handling emojis, stemming, and embedding were evaluated using an appropriate hybrid model. Table 4.2.1 shows the results of Experiment One, in which eliminating emojis from the tweets resulted in poor classification accuracy in R1, R5, R7, and R8. In contrast, encoding emojis into real and emotional meanings improved the classification accuracy in R2, R3, R4, and R6. The highest accuracy, 91.69% was achieved in R3 using the Snowball stemmer and Keras embedding. R2 produced a close result of 90.23% using the ISRI stemmer and Keras embedding. In R4 and R6, the pre-trained AraVec 3.0 had a minimal effect on improving model performance when used with the ISRI and Snowball stemmers, achieving accuracies of 87.32% and 76.09%, respectively. Furthermore, the par plot in Figure 4.2.1 visually represents the results and shows close scores of precision, recall, F1-score, and accuracy, indicating that the explored model performs consistently across all evaluation metrics. This also indicates the effectiveness of the preprocessing steps taken before the training and testing phases. The confusion matrix in Figure 4.2.2 shows that R3 had the lowest number of FP and FN, as reflected by the highest AUC-ROC among all run experiments in Figure 4.2.3 and Table 4.2.1, demonstrating high true positive and true negative rates, as well as the model's ability to distinguish between positive and negative sentiments.

In the second experiment, the performance results in Table 4.3.1 show that keeping the non-Arabic words and removing the punctuation has no positive effect on the model accuracy when using the Keras transformer with the ISRI stemmer and emoji encoding, or the AraVec 3.0 embedding with both used stemmers and emoji encoding in R2, R4, and R6, compared to the performance results in Table 4.2.1. Instead, the accuracy decreased. In contrast, accuracy increased when the non-Arabic words were kept in R3 and R5, and the emojis were translated or eliminated, respectively, using the Snowball stemmer and Keras embedding. This suggests that the combination of the Snowball stemmer and the Keras transformer can better handle the emotions stored in emojis when translating them to Arabic text. Additionally, it can handle words and sentences written in other languages and transform them into insightful vector representations. In contrast, the combination of the ISRI stemmer and the AraVec transformer could not handle non-Arabic words as efficiently as the Keras transformer, resulting in less accurate classification, especially when using the emoji translation technique. This can be explained by the fact that AraVec is a pre-trained model

designed to perform general tasks and aims. It uses Arabic tweets and texts from Wekibidea, which may undergo different preprocessing steps, including the elimination of non-Arabic words. Therefore, the presence of non-Arabic words affects its transformation performance. In contrast, the Keras transformer, a trainable embedding model, updates the vector during the model training based on the same dataset. This helps it provide a better vector representation of the emotions from emojis and non-Arabic words. The best accuracy, 91.85%, was obtained in Experiment Two R3, by keeping the non-Arabic words and removing the punctuation. Non-Arabic sentences often include significant sentiment information that affects the intended meaning of a tweet, leading to a noticeable enhancement in the sentiment prediction accuracy. This is because non-Arabic words are often used in Arabic tweets to highlight the main subject, acting as strong sentiment indicators. For example, a tweet containing the word "love" in the tweet “love أحبك يا قلبي ❤” means " I love you, my heart ❤" in English. The feelings of love referred to in both languages would likely indicate a positive sentiment. Eliminating these non-Arabic tokens would remove important context from the tweet and potentially lead to more wrong predictions. Additionally, the AraVec 3.0 transformer in Experiments Two, R7, and R8 showed slight positive improvement when the non-Arabic words were kept, punctuation was removed, and emojis were removed, compared to Experiments One, R7, and R8. This suggests that eliminating the emojis helps AraVec to provide a better vector representation of tweets by reducing their dimension, since it has a limited output dimension of 100. In contrast, the Keras transformer uses an appropriate output dimension determined by the Keras tuner, which adapts to generate more meaningful full embeddings.

In the third experiment, keeping the non-Arabic tokens and punctuation had the same effect on the model performance for emoji removal and emoji translation experiments as in Experiments One and Two, where the accuracy achieved in the experiments increased dramatically after replacing each emoji with its textual meaning. This was expected during the exploratory data analysis phase, when the word cloud clearly showed the difference in the sentiments of the words in the dataset before and after the emoji encoding step, as shown in Figures 3.3.2 and 3.3.4. Before the emoji translation step, the word cloud in Figure 3.3.2 shows that the high-frequency words are imbalanced between the positive and negative classes, reflecting the low accuracy of experiments that eliminate the emojis. In contrast, after the emoji translation step, the word cloud in Figure 3.3.4 shows a greater number of emotion-related words with high frequency. Additionally, it presents a balance between the number of words with positive and negative words in the word

cloud after translating the emojis to their textual meaning, which indicates the dramatic increase in model accuracy after the emoji encoding step. In the third experiment, the effect of punctuation was also tested by keeping both the punctuation and non-Arabic words to examine their effect on the model accuracy compared to Experiment Two, which kept non-Arabic words and removed the punctuation. The results show that keeping the punctuation negatively affects the model's performance in all runs compared to Experiment Two, especially for R3, which provided the best accuracy in Experiment Two. Meanwhile, the results for R2 and R6 improved. Comparing the results of Experiment One and Two shows the indiscriminate use of punctuation by Twitter users. Thus, eliminating the punctuation creates a more reliable, consistent model. Furthermore, the results obtained in all experiments for all the evaluation metrics demonstrate that the hybrid model is a well-balanced model, as illustrated in Figures 4.2.1, 4.3.1, and 4.4.1.

The results obtained in this thesis using the proposed approach and the ASTC dataset were compared with the results of different studies that used the same Arabic dataset but followed different approaches. Comparisons were made with the two studies in [68] and [49], which are presented in Table 4.5.1 and show differences in study aims, preprocessing steps, and explored models.

Table 4.5.1 Comparison with the state-of-the-art.

Article	**Dataset**	**Model**	**Accuracy**
Our approach	ASTC	CNN-LSTM	**91.85%**
Heterogeneous Ensemble Deep Learning Model for Enhanced Arabic Sentiment Analysis [68]	ASTC	Stacking LR	92.22%
ArabBert-LSTM: improving Arabic sentiment analysis based on a transformer model and Long Short-Term Memory [49]	ASTC	AraBERT-LSTM	93.76%

The proposed model obtained comparable results to those achived in Heterogeneous Ensemble Deep Learning Model for Enhanced Arabic Sentiment Analysis [68] and ArabBert-LSTM: improving Arabic sentiment analysis based on transformer model and Long Short-Term Memory [49]. The former explores the RNN, GRU, and LSTM deep learning models combined with three meta-learners: LR, SVM, and RF, to predict tweet sentiment classes [68]. Also, the emoji Unicode translation and the CBOW word embedding were used before applying the model to provide it with a numerical representation of the text. The approach investigated in [68] aimed to enhance performance in the field of Arabic sentiment analysis. This approach applied several data

preprocessing steps to clean the data by removing non-Arabic letters, single Arabic letters, digits, URLs, symbols, emails, and hashtags [68]. Then, tokenization was performed by splitting the text using spaces [68]. This was followed by eliminating stop words, stemming using the ISRI stemmer, and translating emojis using Unicode. In contrast, [49] investigated several machine and deep learning models to achieve optimal results using the AraBERT-LSTM model, which utilized AraBERTv0 embeddings to advance the ASA field. This study also applied several data preprocessing steps to prepare the data before training and testing the model [49]. These steps included removing URLs, hashtags, punctuation, user mentions, stop words, and commercial and non-Arabic tweets, as well as using tokenization and ISRI stemming [49]. However, the approaches in [64] and [49] do not mention how to handle the more than 18k duplicate rows, which may affect sentiment metric skewing. Moreover, the study in [49] does not mention how to handle emojis.

In contrast, the proposed approach explores the CNN-LSTM hybrid model and finds the effect of various data preprocessing steps on achieving comparable performance results. It also highlights the effects of translating emojis into their emotional and real meanings, as well as the effects of non-Arabic words, punctuation, light and heavy stemmers, and trainable and pre-trained transformers. Furthermore, this study focused on collecting emoji meanings from websites specializing in emoji meanings used on social networks. An important step that the previous two studies did not take was deleting duplicate rows to achieve true accuracy for the machine learning model before examining different preprocessing conditions. These conditions revealed the compatibility between the Keras transformer, Snowball stemmer, and CNN-LSTM model. Including non-Arabic words improved the model's performance, while keeping the punctuation had a negative effect. Furthermore, both our approach and the study in [68] proposed the role of utilizing emoji meanings to enrich the sentiment of Arabic tweets, achieving accuracies of 92.22% and 91.85%, respectively. Our approach provides a comparison of results when eliminating or encoding emojis, validating the emoji meanings in the collected Emoji Meaning dataset.

The lower accuracy achieved in this study using the CNN-LSTM comparison to transformer-based approaches, such as ArabBERT-LSTM and heterogeneous ensemble deep learning models, can be explained by several reasons. These include the fundamental architectural differences and the preprocessing steps followed, like the way of handling the emojis and the duplicated rows. Transformer-based models leverage pretraining on extensive Arabic corpora, enabling them to

comprehend complex, contextualized representations and semantic meanings. Additionally, heterogeneous ensemble models combine multiple complementary learners, making them outperform single architectures. This ensemble strategy reduces model bias and variance and enhances robustness through decision aggregation, which leads to substantially higher computational complexity and training cost. In contrast, the proposed CNN–LSTM framework, which uses convolutional filters to extract local features and LSTM units to extract long-term dependencies, may not fully represent global contextual interactions. Despite this, the CNN-LSTM model remains highly relevant in scenarios where computational efficiency and domain-specific preprocessing are critical. Therefore, the hybrid CNN–LSTM model is a practical and efficient alternative that achieves competitive performance without the heavy resources required by ensemble or transformer-based models.

The findings of this thesis have several practical implications for both industry and academic research. The comparable and high accuracy achieved under different data preprocessing conditions, including the use of emoji translation, demonstrates the potential to improve real-world sentiment monitoring systems used in business, healthcare, marketing, and government institutions. Furthermore, the creation of the Emoji Meaning dataset provides a valuable resource for Arabic sentiment analysis and future NLP research. Thus, the proposed dataset and modeling pipeline offer economic and social benefits by achieving more accurate sentiment classification and creating precise social media monitoring and business analytics applications.

Chapter 5 Conclusion and Future Work

5.1. Conclusion

This thesis investigated the role of data preprocessing in ASA, focusing on enhancing the model with the appropriate data preprocessing methods after developing a suitable CNN-LSTM architecture. Sentiment analysis of Arabic text presents several challenges due to the language's rich morphology, orthographic variations, code-switching, lack of resources, short and noisy texts, domain dependency, diglossia, and semantic ambiguity. Overcoming these challenges requires more attention to Arabic sentiment analysis. Several data preprocessing steps were followed to address these challenges due to their importance in building accurate and reliable models. These steps include static processes, such as removing duplicate rows, hashtags, diacritics, numbers, and stop words, as well as tokenization. They also include variable processes to determine the best way to handle non-Arabic words, punctuation, and emojis, as well as to find the best stemming techniques and embedding transformer models. The fundamental aim of this thesis was to investigate the importance of replacing each emoji with its real and emotional meaning and finding the best combination of preprocessing conditions to enhance Arabic sentiment analysis. Additionally, the thesis aimed to investigate the effect of punctuation and non-Arabic words in ASA.

The thesis's findings and results clearly demonstrate the critical role of data preprocessing in enhancing the model performance in the field of Arabic sentiment analysis. The results of all three experiments showed a significant improvement in model performance across the key evaluation metrics, including accuracy, precision, recall, and F1-score. The main effect on model performance was **based on the emoji handling technique used**; the highest scores were achieved in R2, R3, R4, and R6 when translating the emojis, and the lowest scores were achieved in R1, R5, R7, and R8 when eliminating the emojis. This is likely because social media users tend to enrich their tweets and posts with emojis to indicate real emotions. Additionally, keeping non-Arabic words while removing punctuation and using the Keras transformer and Snowball stemmer proposed the optimal set of preprocessing combinations, obtaining the highest accuracy of 91.85% with the CNN-LSTM model. The integration of the CNN-LSTM model with emoji encoding is a key novel aspect of this thesis, which investigates an underexplored aspect of Arabic sentiment analysis,

specifically using the ASTC dataset. Furthermore, this approach clearly highlights the relationship between the model performance and data preparation, emphasizing the importance of data preprocessing for successful Arabic sentiment analysis models.

This thesis proves that data preprocessing is not a trivial step but an important part of creating a high-performance Arabic sentiment analysis model. By investigating the effect of data preprocessing, this thesis shows how following the right approaches in preparing the data helps overcome the complexity of the Arabic language to build reliable and accurate models. Beyond the direct results, this thesis encourages researchers and developers to pay more attention to data preprocessing.

5.2. Future Work & Recommendations

Based on the outcomes of this thesis and the promising improvements in Arabic sentiment analysis achieved through advanced data preprocessing and emoji encoding techniques, there are several avenues for future work that can be investigated. First, this study can be expanded by integrating these steps with other datasets collected from different domains to make this approach more reliable and generalizable. Second, explore pre-trained transformers such as AraBERT, Glove, and MARBERT, which may offer a more accurate representation of emotions within Arabic texts. Their performance could then be compared against that of the currently utilized transformers, potentially uncovering more efficient ASA models. Third, exploring explainable AI techniques could help interpret how data preprocessing affects model decisions. Additionally, future work could investigate the role of data preprocessing in improving large language model performance. Finally, future work could examine the impact of incorporating insights from speech synthesis, as discussed in [136], into text-based sentiment prediction models. This could result in the development of hybrid models that can understand both Arabic text and speech data.

Publications

The following paper has been published as part of the work presented in this thesis:

- 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. Modelling, 5(4), 1469-1489. https://doi.org/10.3390/modelling5040076

References

A. Diwali, K. Saeedi, K. Dashtipour, M. Gogate, E. Cambria, and A. Hussain, “Sentiment Analysis Meets Explainable Artificial Intelligence: A Survey on Explainable Sentiment Analysis,” *IEEE Transactions on Affective Computing*, 2023, doi: 10.1109/TAFFC.2023.3296373.

B. Saberi and S. Saad, “Sentiment analysis or opinion mining: A review,” *International Journal of Advanced Science Engineering Information Technology*, vol. 7, no. 5, 2017.

M. F. Abdelfattah, M. W. Fakhr, and M. A. Rizka, “ArSentBERT: fine-tuned bidirectional encoder representations from transformers model for Arabic sentiment classification,” *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 1196–1202, Apr. 2023, doi: 10.11591/EEI.V12I2.3914.

A. Alshehri, “Temporal study of the social conversation on Twitter (X): The case of the COVID-19 pandemic,” *South African Journal of Information Management*, vol. 27, no. 1, Jul. 2025, doi: 10.4102/SAJIM.V27I1.2001.

A. Mohammed and R. Kora, “Deep learning approaches for Arabic sentiment analysis,” *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 1–12, Dec. 2019, doi: 10.1007/S13278-019-0596-4/METRICS.

Y. Abdelwahab, M. Kholief, and A. A. H. Sedky, “Justifying Arabic Text Sentiment Analysis Using Explainable AI (XAI): LASIK Surgeries Case Study,” *Information 2022, Vol. 13, Page 536*, vol. 13, no. 11, p. 536, Nov. 2022, doi: 10.3390/INFO13110536.

H. H. Hussein and A. Lakizadeh, “A systematic assessment of sentiment analysis models on iraqi dialect-based texts,” *Systems and Soft Computing*, vol. 7, p. 200203, Dec. 2025, doi: 10.1016/J.SASC.2025.200203.

A. A. Al Shamsi and S. Abdallah, “Ensemble Stacking Model for Sentiment Analysis of Emirati and Arabic Dialects,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101691, Sep. 2023, doi: 10.1016/J.JKSUCI.2023.101691.

O. Oueslati, E. Cambria, M. Ben HajHmida, and H. Ounelli, “A review of sentiment analysis research in Arabic language,” *Future Generation Computer Systems*, vol. 112, pp. 408–430, Nov. 2020, doi: 10.1016/J.FUTURE.2020.05.034.

A. Elnagar, O. Einea, and L. Lulu, “Comparative study of sentiment classification for automated translated Latin reviews into Arabic,” *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, vol. 2017-October, pp. 443–448, Mar. 2018, doi: 10.1109/AICCSA.2017.82.

A. Laggoun, C. Zakaria, K. Smaïli, S. Kamel, Z. Chahnez, and S. Kamel, “Knowledge Distillation for Efficient Algerian Dialect Processing: Training Compact BERT Models

with DziriBERT," *7th International Conference on Advances in Signal Processing and Artificial Intelligence*, Apr. 2025, doi: 10.34894/VQ1DJA.

V. N. Gudivada and S. Albahli, "An Advanced Natural Language Processing Framework for Arabic Named Entity Recognition: A Novel Approach to Handling Morphological Richness and Nested Entities," *Applied Sciences 2025, Vol. 15, Page 3073*, vol. 15, no. 6, p. 3073, Mar. 2025, doi: 10.3390/APP15063073.

S. Tibi, L. Fitton, and A. L. McIlraith, "The development of a measure of orthographic knowledge in the Arabic language: A psychometric evaluation," *Applied Psycholinguistics*, vol. 42, no. 3, pp. 739–762, May 2021, doi: 10.1017/S0142716421000035.

I. Hamed, C. Sabty, S. Abdennadher, N. T. Vu, T. Solorio, and N. Habash, "A Survey of Code-switched Arabic NLP: Progress, Challenges, and Future Directions," *arXiv preprint arXiv:2501.13419*, Jan. 2025, Accessed: May 01, 2025. [Online]. Available: https://arxiv.org/pdf/2501.13419

S. M. Alzanin, A. M. Azmi, and H. A. Aboalsamh, "Short text classification for Arabic social media tweets," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6595–6604, Oct. 2022, doi: 10.1016/j.jksuci.2022.03.020.

B. Liu, Sentiment Analysis and Opinion Mining. *Springer Nature*, 2022, Accessed: May 05, 2025. [Online]. Available: https://books.google.ps/books?hl=en&lr=&id=xYhyEAAAQBAJ&oi=fnd&pg=PP1&ots=rlVALGQ3Cv&sig=18QsBXWSvQ7ax6pQwOUvGvI3uBo&redir_esc=y#v=onepage&q&f=false

Y. Alqahtani, N. Al-Twairesh, and A. Alsanad, "A Comparative Study of Effective Domain Adaptation Approaches for Arabic Sentiment Classification," *Applied Sciences 2023, Vol. 13, Page 1387*, vol. 13, no. 3, p. 1387, Jan. 2023, doi: 10.3390/APP13031387.

N. Essa, M. M. El-Gayar, and E. M. El-Daydamony, "Enhanced model for abstractive Arabic text summarization using natural language generation and named entity recognition," *Neural Computing and Applications*, vol. 37, no. 10, pp. 7279–7301, Apr. 2025, doi: 10.1007/S00521-024-10949-X/TABLES/9.

S. Al-Azani and E. S. M. El-Alfy, "Combining emojis with Arabic textual features for sentiment classification," *2018 9th International Conference on Information and Communication Systems, ICICS 2018*, vol. 2018-January, pp. 139–144, May 2018, doi: 10.1109/IACS.2018.8355456.

P. K. Novak, J. Smailović, B. Sluban, and I. Mozetič, "Sentiment of Emojis," *PLoS One*, vol. 10, no. 12, p. e0144296, Dec. 2015, doi: 10.1371/JOURNAL.PONE.0144296.

N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment Analysis Based on Deep Learning: A Comparative Study," *Electronics (Switzerland)*, vol. 9, no. 3, Jun. 2020, doi: 10.3390/electronics9030483.

M. A. Bin Syed and I. Ahmed, “A CNN-LSTM Architecture for Marine Vessel Track Association Using Automatic Identification System (AIS) Data,” *Sensors 2023, Vol. 23, Page 6400*, vol. 23, no. 14, p. 6400, Jul. 2023, doi: 10.3390/S23146400.

F. Hu, Q. Yang, J. Yang, Z. Luo, J. Shao, and G. Wang, “Incorporating multiple grid-based data in CNN-LSTM hybrid model for daily runoff prediction in the source region of the Yellow River Basin,” *Journal of Hydrology: Regional Studies*, vol. 51, p. 101652, Feb. 2024, doi: 10.1016/J.EJRH.2023.101652.

R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/S13244-018-0639-9/FIGURES/15.

A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, “A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages,” *Future Internet 2020, Vol. 12, Page 156*, vol. 12, no. 9, p. 156, Sep. 2020, doi: 10.3390/FI12090156.

C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, “A C-LSTM Neural Network for Text Classification,” *arXiv preprint arXiv:1511.08630*, Nov. 2015, Accessed: May 10, 2025. [Online]. Available: https://arxiv.org/pdf/1511.08630

M. Soleymani, D. Garcia, B. Jou, B. Schuller, S. F. Chang, and M. Pantic, “A survey of multimodal sentiment analysis,” *Image and Vision Computing*, vol. 65, pp. 3–14, Sep. 2017, doi: 10.1016/J.IMAVIS.2017.08.003.

W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, “User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models,” *Applied Soft Computing*, vol. 94, p. 106435, Sep. 2020, doi: 10.1016/J.ASOC.2020.106435.

R. M. Duwairi and I. Qarqaz, “Arabic sentiment analysis using supervised classification,” *Proceedings - 2014 International Conference on Future Internet of Things and Cloud, FiCloud 2014*, pp. 579–583, Dec. 2014, doi: 10.1109/FICLOUD.2014.100.

A. Radman and R. Duwairi, “Towards a robust deep learning framework for Arabic sentiment analysis,” *Natural Language Processing*, vol. 31, no. 2, pp. 500–534, Mar. 2025, doi: 10.1017/NLP.2024.35.

A. H. Ombabi, W. Ouarda, and A. M. Alimi, “Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks,” *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 1–13, Dec. 2020, doi: 10.1007/S13278-020-00668-1/METRICS.

M. E. M. Abo, R. G. Raj, and A. Qazi, “A Review on Arabic sentiment analysis: State-of-The-Art, taxonomy and open research challenges,” *IEEE Access*, vol. 7, pp. 162008–162024, 2019, doi: 10.1109/ACCESS.2019.2951530.

M. Kumar, L. Khan, and H.-T. Chang, “Evolving techniques in sentiment analysis: a comprehensive review,” *PeerJ Computer Science*, vol. 11, p. e2592, Jan. 2025, doi: 10.7717/PEERJ-CS.2592.

O. Alsemaree, A. S. Alam, S. S. Gill, and S. Uhlig, “Sentiment analysis of Arabic social media texts: A machine learning approach to deciphering customer perceptions,” *Heliyon*, vol. 10, no. 9, pp. 2405–8440, May 2024, doi: 10.1016/j.heliyon.2024.e27863.

D. A. Musleh *et al.*, “Arabic Sentiment Analysis of YouTube Comments: NLP-Based Machine Learning Approaches for Content Evaluation,” *Big Data and Cognitive Computing 2023, Vol. 7, Page 127*, vol. 7, no. 3, p. 127, Jul. 2023, doi: 10.3390/BDCC7030127.

E. Dritsas, Z. Nassr, F. Benabbou, N. Sael, and T. Hamim, “Improving Sentiment Analysis Performance on Imbalanced Moroccan Dialect Datasets Using Resample and Feature Extraction Techniques,” *Information 2025, Vol. 16, Page 39*, vol. 16, no. 1, p. 39, Jan. 2025, doi: 10.3390/INFO16010039.

K. M. Alomari, H. M. Elsherif, and K. Shaalan, “Arabic Tweets Sentimental Analysis Using Machine Learning,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10350 LNCS, pp. 602–610, 2017, doi: 10.1007/978-3-319-60042-0_66.

M. Almufti, M. Elamine, and L. Hadrich Belguith, “Comparative analysis of machine and deep learning algorithms for semantic analysis in Iraqi dialect,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 37, no. 2, pp. 1225–1233, Feb. 2025, doi: 10.11591/IJEECS.V37.I2.PP1225-1233.

N. E. Aoumeur, Z. Li, and E. M. Alshari, “Improving the Polarity of Text through word2vec Embedding for Primary Classical Arabic Sentiment Analysis,” *Neural processing letters*, vol. 55, no. 3, pp. 2249–2264, Jun. 2023, doi: 10.1007/S11063-022-11111-1/FIGURES/7.

A. H. Allam, H. M. Abdallah, E. Amer, and H. A. Nayel, “Machine Learning-Based Model for Sentiment and Sarcasm Detection,” Proceedings of the Sixth Arabic Natural Language Processing Workshop, 2021. Accessed: Sep. 07, 2023. [Online]. Available: https://aclanthology.org/2021.wanlp-1.51

S. F. Sabbeh and H. A. Fasihuddin, “A Comparative Analysis of Word Embedding and Deep Learning for Arabic Sentiment Classification,” *Electronics 2023, Vol. 12, Page 1425*, vol. 12, no. 6, p. 1425, Mar. 2023, doi: 10.3390/ELECTRONICS12061425.

N. Elhassan *et al.*, “Arabic Sentiment Analysis Based on Word Embeddings and Deep Learning,” *Computers 2023, Vol. 12, Page 126*, vol. 12, no. 6, p. 126, Jun. 2023, doi: 10.3390/COMPUTERS12060126.

H. Gharaibeh *et al.*, “Arabic sentiment analysis of Monkeypox using deep neural network and optimized hyperparameters of machine learning algorithms,” *Social Network Analysis*

and Mining, vol. 14, no. 1, pp. 1–18, Dec. 2024, doi: 10.1007/S13278-023-01188-4/METRICS.

M. Mhamed, R. Sutcliffe, and J. Feng, "Benchmark Arabic news posts and analyzes Arabic sentiment through RMuBERT and SSL with AMCFFL technique," *Egyptian Informatics Journal*, vol. 29, p. 100601, Mar. 2025, doi: 10.1016/J.EIJ.2024.100601.

M. Ramzy and B. Ibrahim, "User satisfaction with Arabic COVID-19 apps: Sentiment analysis of users' reviews using machine learning techniques," *Information Processing & Management*, vol. 61, no. 3, p. 103644, May 2024, doi: 10.1016/J.IPM.2024.103644.

A. Wadhawan, "AraBERT and Farasa Segmentation Based Approach For Sarcasm and Sentiment Detection in Arabic Tweets," *arXiv preprint arXiv:2103.01679*, Mar. 2021, Accessed: Sep. 07, 2023. [Online]. Available: https://arxiv.org/abs/2103.01679v1

H. M. Alghamdi, "Unveiling Sentiments: A Comprehensive Analysis of Arabic Hajj-Related Tweets from 2017–2022 Utilizing Advanced AI Models," *Big Data and Cognitive Computing 2024, Vol. 8, Page 5*, vol. 8, no. 1, p. 5, Jan. 2024, doi: 10.3390/BDCC8010005.

M. S. Islam *et al.*, "'Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach,'" *Artificial Intelligence Review*, vol. 57, no. 3, pp. 1–79, Mar. 2024, doi: 10.1007/S10462-023-10651-9/FIGURES/13.

W. Alosaimi *et al.*, "ArabBert-LSTM: improving Arabic sentiment analysis based on transformer model and Long Short-Term Memory," *Frontiers in Artificial Intelligence*, vol. 7, p. 1408845, Jul. 2024, doi: 10.3389/FRAI.2024.1408845/BIBTEX.

S. N. Alyami and S. O. Olatunji, "Application of Support Vector Machine for Arabic Sentiment Classification Using Twitter-Based Dataset," *Journal of Information & Knowledge Management*, vol. 19, no. 1, Apr. 2020, doi: 10.1142/S0219649220400183.

"Arabic Sentiment Twitter Corpus," *Kaggle*, Accessed: Mar. 31, 2024. [Online]. Available: https://www.kaggle.com/datasets/mksaad/arabic-sentiment-twitter-corpus/data?select=arabic_tweets

A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Arabic language sentiment analysis on health services," *2017 1st international workshop on arabic script analysis and recognition*, pp. 114–118, Oct. 2017, doi: 10.1109/ASAR.2017.8067771.

A. Alwehaibi, M. Bikdash, M. Albogmi, and K. Roy, "A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 6140–6149, Sep. 2022, doi: 10.1016/J.JKSUCI.2021.07.011.

N. Al-Twairesh, H. Al-Khalifa, A. Al-Salman, and Y. Al-Ohali, "AraSenTi-Tweet: A Corpus for Arabic Sentiment Analysis of Saudi Tweets," *Procedia Computer Science*, vol. 117, pp. 63–72, Jan. 2017, doi: 10.1016/J.PROCS.2017.10.094.

D. M. A. Bakhit, L. Nderu, and A. Ngunyi, “A hybrid neural network model based on transfer learning for Arabic sentiment analysis of customer satisfaction,” *Engineering Reports*, vol. 6, no. 10, p. e12874, Oct. 2024, doi: 10.1002/ENG2.12874.

“Arabic 100k Reviews,” *Kaggle*, Accessed: Feb. 07, 2025. [Online]. Available: https://www.kaggle.com/datasets/abedkhooli/arabic-100k-reviews

M. Nabil, M. Aly, and A. Atiya, “Astd: Arabic sentiment tweets dataset,” *Proceedings of the 2015 conference on empirical methods in natural, 2015*, pp. 17–21, 2015, Accessed: Apr. 29, 2024. [Online]. Available: https://aclanthology.org/D15-1299.pdf

L. Almuqren and A. Cristea, “AraCust: a Saudi Telecom Tweets corpus for sentiment analysis,” *PeerJ Computer Science*, vol. 7, pp. 1–30, May 2021, doi: 10.7717/PEERJ-CS.510/SUPP-2.

N. Hicham, H. Nassera, and S. Karim, “Enhancing Arabic E-Commerce Review Sentiment Analysis Using a hybrid Deep Learning Model and FastText word embedding,” *EAI Endorsed Transactions on Internet of Things*, vol. 10, 2024, doi: 10.4108/EETIOT.4601.

Y. Zahidi, Y. Al-Amrani, and Y. El Younoussi, “Deep Learning CNN–LSTM Hybrid Approach for Arabic Sentiment Analysis Using Word Embedding Models,” *International Journal of Modern Education and Computer Science*, vol. 17, no. 1, pp. 72–90, Feb. 2025, doi: 10.5815/IJMECS.2025.01.06.

A. M. Alayba and V. Palade, “Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 9710–9722, Nov. 2022, doi: 10.1016/J.JKSUCI.2021.12.004.

N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub, “Arabic sentiment analysis: Lexicon-based and corpus-based,” *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2013*, 2013, doi: 10.1109/AEECT.2013.6716448.

A. Hengle, A. Kshirsagar, S. Desai, and M. Marathe, “Combining Context-Free and Contextualized Representations for Arabic Sarcasm Detection and Sentiment Identification,” *WANLP 2021 - 6th Arabic Natural Language Processing Workshop, Proceedings of the Workshop*, pp. 357–363, Mar. 2021, Accessed: Sep. 07, 2023. [Online]. Available: https://arxiv.org/abs/2103.05683v1

“ArSarcasm-V2,” *Kaggle*, Accessed: Jul. 29, 2023. [Online]. Available: https://www.kaggle.com/datasets/abraralotaibi00/arsarcasm-v2?select=testing_data.csv

A. A. Jalil and A. H. Aliwy, “Classification of Arabic Social Media Texts Based on a Deep Learning Multi-Tasks Model,” *Al-Bahir Journal for Engineering and Pure Sciences*, vol. 2, no. 2, p. 12, May 2023, doi: 10.55810/2312-5721.1030.

S. Al-Azani and E. S. M. El-Alfy, “Emoji-Based Sentiment Analysis of Arabic Microblogs Using Machine Learning,” *21st Saudi Computer Society National Computer Conference, NCC 2018*, Dec. 2018, doi: 10.1109/NCG.2018.8592970.

A. A. Arifiyanti and E. D. Wahyuni, “Emoji and emoticon in tweet sentiment classification,” *Proceeding - 6th Information Technology International Seminar, ITIS 2020*, pp. 145–150, Oct. 2020, doi: 10.1109/ITIS50118.2020.9320988.

H. Saleh, S. Mostafa, A. Alharbi, S. El-Sappagh, and T. Alkhalifah, “Heterogeneous Ensemble Deep Learning Model for Enhanced Arabic Sentiment Analysis,” *Sensors 2022, Vol. 22, Page 3707*, vol. 22, no. 10, p. 3707, May 2022, doi: 10.3390/S22103707.

N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub, “Arabic sentiment analysis: Lexicon-based and corpus-based,” *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2013*, 2013, doi: 10.1109/AEECT.2013.6716448.

A. Surikov and E. Egorova, “Alternative method sentiment analysis using emojis and emoticons,” *Procedia Computer Science*, vol. 178, pp. 182–193, Jan. 2020, doi: 10.1016/J.PROCS.2020.11.020.

A. Rogers, A. Romanov, A. Rumshisky, S. Volkova, M. Gronas, and A. Gribov, “RuSentiment: An Enriched Sentiment Analysis Dataset for Social Media in Russian,” *Proceedings of the 27th international conference on computational linguistics*, pp. 755-763, 2018. Accessed: Apr. 21, 2025. [Online]. Available: https://aclanthology.org/C18-1064/

E. A. H. Khalil, E. M. F. El Houby, and H. K. Mohamed, “Deep learning for emotion analysis in Arabic tweets,” *Journal of Big Data*, vol. 8, no. 1, pp. 1–15, Dec. 2021, doi: 10.1186/S40537-021-00523-W/TABLES/7.

S. M. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, “SemEval-2018 Task 1: Affect in Tweets,” *NAACL HLT 2018 - International Workshop on Semantic Evaluation, SemEval 2018 - Proceedings of the 12th Workshop*, pp. 1–17, 2018, doi: 10.18653/V1/S18-1001.

S. Al-Azani and E. S. El-Alfy, “Emojis-based sentiment classification of Arabic microblogs using deep recurrent neural networks,” *2018 International Conference on Computing Sciences and Engineering, ICCSE 2018 - Proceedings*, pp. 1–6, Jun. 2018, doi: 10.1109/ICCSE1.2018.8374211.

S. M. Mohammad, M. Salameh, and S. Kiritchenko, “How Translation Alters Sentiment,” *Journal of Artificial Intelligence Research*, vol. 55, pp. 95–130, Jan. 2016, doi: 10.1613/JAIR.4787.

M. Alfreihat, O. S. Almousa, Y. Tashtoush, A. Alsobeh, K. Mansour, and H. Migdady, “Emo-SL Framework: Emoji Sentiment Lexicon Using Text-Based Features and Machine Learning for Sentiment Analysis,” *IEEE Access*, vol. 12, pp. 81793–81812, 2024, doi: 10.1109/ACCESS.2024.3382836.

A. Mourad and K. Darwish, “Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs,” *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pp. 55–64, 2013, Accessed: Apr. 23, 2025. [Online]. Available: http://en.wikipedia.org/wiki/Varieties_

Y. Chen, Q. You, J. Yuan, and J. Luo, “Twitter sentiment analysis via bi-sense emoji embedding and attention-based LSTM,” *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference*, pp. 117–125, Oct. 2018, doi: 10.1145/3240508.3240533.

W. Bunyatisai, S. Plubin, K. Jiamwattanapong, and B. Plubin, “Improving Thai sentiment analysis accuracy with emoji classification by deep learning and stacking models: A case study of hotel reviews,” *Pakistan Journal of Life and Social Sciences*, vol. 23, no. 1, pp. 65–77, 2025, doi: 10.57239/PJLSS-2025-23.1.007.

P. Dadure, U. Dehradun, A. Dixit, K. Tewatia, N. Paliwal, and A. Malla, “Sentiment Analysis of Arabic Tweets Using Large Language Models,” *Proceedings of the 1st Workshop on NLP for Languages Using Arabic Script*, Accessed: Feb. 02, 2025. [Online]. Available: https://aclanthology.org/2025.abjadnlp-1.10/

“EmojiGuide,” Accessed: Apr. 09, 2024. [Online]. Available: https://ar.emojiguide.com/

“EmojiAll,” Accessed: Apr. 09, 2024. [Online]. Available: https://www.emojiall.com/ar

“Symbol Planet,” Accessed: Apr. 09, 2024. [Online]. Available: https://symbolplanet.com/

“wikiHow,” Accessed: Apr. 09, 2024. [Online]. Available: https://www.wikihow.com/Main-Page

M. Sadiku, A. E. Shadare, S. M. Musa, C. M. Akujuobi, and R. Perry, “DATA VISUALIZATION,” *International Journal of Engineering Research And Advanced Technology (IJERAT)*, Accessed: Apr. 10, 2023. [Online]. Available: https://www.researchgate.net/publication/311597028_DATA_VISUALIZATION

S. Rawat, A. Rawat, D. Kumar, and A. S. Sabitha, “Application of machine learning and data visualization techniques for decision support in the insurance sector,” *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100012, Nov. 2021, doi: 10.1016/J.JJIMEI.2021.100012.

W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li, “An Ensemble Random Forest Algorithm for Insurance Big Data Analysis,” *IEEE Access*, vol. 5, pp. 16568–16575, 2017, Accessed: Apr. 22, 2023. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8007210

A. Yacob, N. E. Ghazali, and F. M. Hassan, “Sentiment Analysis of ChatGPT Using the KNN Algorithm and K-Fold Cross-Validation Optimization of the K Value,” *JICO: International Journal of Informatics and Computing*, vol. 1, no. 2, pp. 48–55, 2024, Accessed: May 23, 2025. [Online]. Available: https://www.researchgate.net/publication/389515999_Sentiment_Analysis_of_ChatGPT_

Using_the_KNN_Algorithm_and_K-Fold_Cross-Validation_Optimization_of_the_K_Value

Z. Ma, A. Sun, Q. Yuan, and G. Cong, "Tagging your tweets: A probabilistic modeling of hashtag annotation in twitter," *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 999–1008, Nov. 2014, doi: 10.1145/2661829.2661903.

L. Yang, T. Sun, M. Zhang, and Q. Mei, "We know what @you #tag: Does the dual role affect hashtag adoption?," *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web*, pp. 261–270, 2012, doi: 10.1145/2187836.2187872.

D. Bamman and N. A. Smith, "Contextualized Sarcasm Detection on Twitter," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 9, no. 1, pp. 574–577, 2015, doi: 10.1609/ICWSM.V9I1.14655.

R. M. Duwairi, R. Marji, N. Sha'Ban, and S. Rushaidat, "Sentiment analysis in arabic tweets," *2014 5th International Conference on Information and Communication Systems, ICICS 2014*, 2014, doi: 10.1109/IACS.2014.6841964.

N. Khalid Bolbol and A. Y. Maghari, "Sentiment analysis of arabic tweets using supervised machine learning," *Proceedings - 2020 International Conference on Promising Electronic Technologies, ICPET 2020*, pp. 89–93, Dec. 2020, doi: 10.1109/ICPET51420.2020.00025.

N. Khamphakdee and P. Seresangtakul, "An Efficient Deep Learning for Thai Sentiment Analysis," *Data 2023, Vol. 8, Page 90*, vol. 8, no. 5, p. 90, May 2023, doi: 10.3390/DATA8050090.

Y. Albalawi, J. Buckley, and N. S. Nikolov, "Investigating the impact of pre-processing techniques and pre-trained word embeddings in detecting Arabic health information on social media," *Journal of big Data*, vol. 8, no. 1, pp. 1–29, Dec. 2021, doi: 10.1186/S40537-021-00488-W/TABLES/2.

M. Al-Helalat, "ENHANCED ARABIC INFORMATION RETRIEVAL FOR INFORMED DECISION-MAKING: EMPOWERING POLITICAL SEARCH," *INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)*, vol. 03, pp. 232–240, 2023, Accessed: May 10, 2024. [Online]. Available: https://www.ijprems.com/uploadedfiles/paper/issue_7_july_2023/31816/final/fin_ijprems1689480149.pdf

I. A. El-Khair, "Effects of Stop Words Elimination for Arabic Information Retrieval: A Comparative Study," *arXiv preprint arXiv:1702.01925*, Feb. 2017, Accessed: May 24, 2025. [Online]. Available: https://arxiv.org/pdf/1702.01925

H. Chouikhi, H. Chniter, and F. Jarray, "Arabic Sentiment Analysis Using BERT Model," *Communications in Computer and Information Science*, vol. 1463, pp. 621–632, 2021, doi: 10.1007/978-3-030-88113-9_50.

S. Kannan, V. Gurusamy, S. Vijayarani, J. Ilamathi, N. Kannan, and V. Gurusamy, "Preprocessing Techniques for Text Mining" *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2014.

R. Van Der Goot, "Where are we Still Split on Tokenization?," *Findings of the Association for Computational Linguistics: EACL*, 2024. Accessed: Apr. 27, 2024. [Online]. Available: https://aclanthology.org/2024.findings-eacl.9

S. Bird, "NLTK: the natural language toolkit," *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, 2006*, pp. 69–72, Accessed: Apr. 27, 2024. [Online]. Available: https://aclanthology.org/P06-4018.pdf

J. Islam, R. E. Mercer, and L. Xiao, "Multi-Channel Convolutional Neural Network for Twitter Emotion and Sentiment Recognition," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 1355–1365, 2019, doi: 10.18653/V1/N19-1137.

F. Barbieri, M. Ballesteros, and H. Saggion, "Are Emojis Predictable?," *arXiv preprint arXiv:1702.07285*, 2017. Accessed: May 25, 2025. [Online]. Available: https://aclanthology.org/E17-2017/

M. Maree, M. Eleyat, S. Rabayah, and M. Belkhatir, "A hybrid composite features based sentence level sentiment analyzer," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 1, pp. 284–294, Mar. 2023, doi: 10.11591/IJAI.V12.I1.PP284-294.

E. Haddi, X. Liu, and Y. Shi, "The Role of Text Pre-processing in Sentiment Analysis," *Procedia computer sciencei*, vol. 17, pp. 26–32, Jan. 2013, doi: 10.1016/J.PROCS.2013.05.005.

H. Alshalabi, S. Tiun, N. Omar, F. N. AL-Aswadi, and K. Ali Alezabi, "Arabic light-based stemmer using new rules," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6635–6642, Oct. 2022, doi: 10.1016/J.JKSUCI.2021.08.017.

M. N. Al-Kabi, S. A. Kazakzeh, B. M. Abu Ata, S. A. Al-Rababah, and I. M. Alsmadi, "A novel root based Arabic stemmer," *Journal of King Saud University - Computer and Information Sciences*, vol. 27, no. 2, pp. 94–103, Apr. 2015, doi: 10.1016/J.JKSUCI.2014.04.001.

K. Taghva, R. Elkhoury, and J. Coombs, "Arabic stemming without a root dictionary," *International Conference on Information Technology: Coding and Computing, ITCC*, vol. 1, pp. 152–157, 2005, doi: 10.1109/ITCC.2005.90.

M. F. Porter, "Snowball: A language for stemming algorithms." 2001, Accessed: May 29, 2025. [Online]. Available: https://snowballstem.org/texts/introduction.html

A. Radwan, M. Amarneh, H. Alawneh, H. I. Ashqar, A. AlSobeh, and A. A. A. R. Magableh, "Predictive Analytics in Mental Health Leveraging LLM Embeddings and

Machine Learning Models for Social Media Analysis," *International Journal of Web Services Research (IJWSR)*, vol. 21, no. 1, pp. 1–22, Jan. 1AD, doi: 10.4018/IJWSR.338222.

A. Gulli and S. Pal, Deep Learning with Keras, *Packt Publishing Ltd*, 2017, Accessed: May 09, 2024. [Online]. Available: https://books.google.ps/books?hl=en&lr=&id=20EwDwAAQBAJ&oi=fnd&pg=PP1&dq=Keras+embedding+to+minimize+the+loss&ots=lJ9A6mgQR3&sig=5FCEyP6VRUtL5pdNO8k9iHHqX3g&redir_esc=y#v=onepage&q=Keras%20embedding%20to%20minimize%20the%20loss&f=false

A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP," *Procedia Computer Science*, vol. 117, pp. 256–265, 2017, doi: 10.1016/J.PROCS.2017.10.117.

J. María, A. Rodríguez, A. Allahim, and A. Cherif, "Advancing Arabic Word Embeddings: A Multi-Corpora Approach with Optimized Hyperparameters and Custom Evaluation," *Applied Sciences 2024, Vol. 14, Page 11104*, vol. 14, no. 23, p. 11104, Nov. 2024, doi: 10.3390/APP142311104.

H. Elzayady, K. M. Badran, and G. I. Salama, "Arabic Opinion Mining Using Combined CNN-LSTM Models," *Article in International Journal of Intelligent Systems and Applications*, vol. 4, pp. 25–36, 2020, doi: 10.5815/ijisa.2020.04.03.

P. M. Sosa, "Twitter Sentiment Analysis using combined LSTM-CNN Models," *Eprint Arxiv*, 2017, pp. 1–9.

A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11015 LNCS, pp. 179–191, 2018, doi: 10.1007/978-3-319-99740-7_12/FIGURES/5.

H. Alawneh, A. Hasasneh, and M. Maree, "On the Utilization of Emoji Encoding and Data Preprocessing with a Combined CNN-LSTM Framework for Arabic Sentiment Analysis," *Modelling, Vol. 5, Pages 1469-1489*, vol. 5, no. 4, pp. 1469–1489, Oct. 2024, doi: 10.3390/MODELLING5040076.

Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

H. Saleh, S. Mostafa, L. A. Gabralla, A. O. Aseeri, and S. El-Sappagh, "Enhanced Arabic Sentiment Analysis Using a Novel Stacking Ensemble of Hybrid and Deep Learning Models," *Applied Sciences 2022, Vol. 12, Page 8967*, vol. 12, no. 18, p. 8967, Sep. 2022, doi: 10.3390/APP12188967.

A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis," *Multimedia Tools and*

Applications, vol. 78, no. 18, pp. 26597–26613, Sep. 2019, doi: 10.1007/S11042-019-07788-7/METRICS.

L. Javier *et al.*, "Deep Sentiment Analysis Using CNN-LSTM Architecture of English and Roman Urdu Text Shared in Social Media," *Applied Sciences 2022, Vol. 12, Page 2694*, vol. 12, no. 5, p. 2694, Mar. 2022, doi: 10.3390/APP12052694.

R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data," *Information Processing & Management*, vol. 58, no. 1, p. 102435, Jan. 2021, doi: 10.1016/J.IPM.2020.102435.

A. Jafar and M. Lee, "Comparative Performance Evaluation of State-of-the-Art Hyperparameter Optimization Frameworks," *Transactions of the Korean Institute of Electrical Engineers*, vol. 72, no. 5, pp. 607–619, May 2023, doi: 10.5370/KIEE.2023.72.5.607.

"KerasTuner." Accessed: Apr. 12, 2024. [Online]. Available: https://keras.io/keras_tuner/

H. Alawneh and A. Hasasneh, "Survival Prediction of Children after Bone Marrow Transplant Using Machine Learning Algorithms," *The International Arab Journal of Information Technology*, vol. 21, no. 3, 2024, doi: 10.34028/iajit/21/3/4.

V. S. Spelmen and R. Porkodi, "A Review on Handling Imbalanced Data," *Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies, ICCTCT 2018*, Nov. 2018, doi: 10.1109/ICCTCT.2018.8551020.

M. Ghanem *et al.*, "Limitations in Evaluating Machine Learning Models for Imbalanced Binary Outcome Classification in Spine Surgery: A Systematic Review," *Brain Sciences*, vol. 13, no. 12, p. 1723, Dec. 2023, doi: 10.3390/BRAINSCI13121723/S1.

A. Karanikola, G. Davrazos, C. M. Liapis, and S. Kotsiantis, "Financial sentiment analysis: Classic methods vs. deep learning models," *Intelligent Decision Technologies*, vol. 17, no. 4, pp. 893–915, Nov. 2023, doi: 10.3233/IDT-230478/ASSET/47180F72-F014-4CFE-8264-8C341C1500AD/ASSETS/IMAGES/LARGE/10.3233_IDT-230478-FIG5.JPG.

M. A. Islam and I. E. Iacob, "Manuscripts Character Recognition Using Machine Learning and Deep Learning," *Modelling 2023, Vol. 4, Pages 168-188*, vol. 4, no. 2, pp. 168–188, Apr. 2023, doi: 10.3390/MODELLING4020010.

O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, vol. 14, no. 1, pp. 1–14, Dec. 2024, doi: 10.1038/S41598-024-56706-X;SUBJMETA=117,531,639,705;KWRD=COMPUTER+SCIENCE,STATISTICS.

S. Sathyanarayanan, "Confusion Matrix-Based Performance Evaluation Metrics," *African Journal of Biomedical Research*, pp. 4023–4031, Nov. 2024, doi: 10.53555/AJBR.V27I4S.4345.

D. Tiwari, B. Nagpal, B. S. Bhati, A. Mishra, and M. Kumar, “A systematic review of social network sentiment analysis with comparative study of ensemble-based techniques,” *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13407–13461, Nov. 2023, doi: 10.1007/S10462-023-10472-W/TABLES/22.

T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets.,” *PLoS One*, vol. 10, no. 3, pp. e0118432–e0118432, Mar. 2015, doi: 10.1371/JOURNAL.PONE.0118432.

“File:Roc curve.svg,” Wikimedia Commons, Accessed: Jun. 28, 2025. [Online]. Available: https://commons.wikimedia.org/wiki/File:Roc_curve.svg

Q. Li, “Overview of Data Visualization,” *Embodying Data*, pp. 17–47, 2020, doi: 10.1007/978-981-15-5069-0_2.

M. S. Al-Radhi, O. Abdo, T. G. Csapó, S. Abdou, G. Németh, and M. Fashal, “A continuous vocoder for statistical parametric speech synthesis and its evaluation using an audio-visual phonetically annotated Arabic corpus,” *Computer Speech & Language*, vol. 60, p. 101025, Mar. 2020, doi: 10.1016/J.CSL.2019.101025.

الملخص

غالبًا ما يعبر مستخدمو وسائل التواصل الاجتماعي عن مشاعرهم وأفكارهم وآرائهم من خلال النصوص التي ينشرونها في المنشورات والتغريدات، والتي يمكن استخدامها لتحديد قطبية النص على أنه إيجابي أو سلبي - وهي عملية تُعرف باسم تحليل المشاعر. أصبح تحليل المشاعر أمرًا بالغ الأهمية في العديد من المجالات الواقعية، بما في ذلك السياسة والسياحة والتجارة الإلكترونية والتعليم والصحة. ومع ذلك، على الرغم من أن أساليب تحليل المشاعر تعمل بشكل جيد مع النصوص الإنجليزية، فإنها تواجه عوائق ملحوظة عند التعامل مع النصوص العربية. تشكل التعقيدات الصرفية المتأصلة في اللغة العربية تحديات لبناء نماذج قوية، مما يجعل من الصعب فهم المشاعر العامة واتخاذ قرارات مستنيرة في وقت لاحق.

استجابة لهذه التحديات، يتم استخدام تقنيات فعالة لمعالجة البيانات المسبقة والتعلم العميق للتغلب على تعقيد اللغة العربية وتقديم تنبؤات ثاقبة للمشاعر. تقيّم هذه الأطروحة إطار عمل مشترك بين Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) مع تقنيات مختلفة لمعالجة البيانات المسبقة لتحليل المشاعر العربية باستخدام مجموعة بيانات Arabic Sentiment Twitter Corpus (ASTC). أجريت ثلاث تجارب بثمانية تكوينات مختلفة للمعالجة المسبقة لتقييم تأثير المعالجة المسبقة للبيانات على تحليل المشاعر العربية، وتحديدًا تأثير ترميز وترجمة الرموز التعبيرية إلى معانيها الحرفية والعاطفية. تم جمع معاني الرموز التعبيرية من أربعة مواقع إلكترونية متخصصة في تعريف معاني الرموز التعبيرية في وسائل التواصل الاجتماعي، مما أدى إلى إنشاء مجموعة بيانات جديدة لمعاني الرموز التعبيرية تسمى مجموعة بيانات "Emoji Meaning ". علاوة على ذلك، تم تحسين معلمات CNN-LSTM باستخدام Keras Tuner خلال عملية five-fold cross-validation.

حصل النموذج المقترح مع ترجمة الرموز التعبيرية إلى نص عربي على أعلى معدل دقة (91.85٪) من خلال الاحتفاظ بالكلمات غير العربية وإزالة علامات الترقيم واستخدام أداة Snowball stemmer واستخدام Keras embedding. ينتج عن هذا النهج نتائج تنافسية مقارنة بالأساليب الحديثة الأخرى، ويثبت أن ترميز الرموز التعبيرية يثري النص من خلال عكس المشاعر بدقة، ويبحث في تأثير المعالجة المسبقة للبيانات على أداء النموذج. وهذا يسمح

للنموذج الهجين بتحقيق نتائج مماثلة للدراسات الأخرى التي تستخدم نفس مجموعة بيانات ASTC، وبالتالي تحسين دقة تحليل المشاعر.