Ain Shams University
Faculty of Computer & Information Sciences
Information Systems Department

Arabic Sentiment Analysis

This documentation submitted as required for the degree of bachelor's in computer and information sciences

By

Ali Abdallah Mohamed Abdallah [Information Sciences]

Mohamed Ali Abdelaziz Abdelaziz [Information Sciences]

Karima Sobhi Abo Al-ala Abd Elkhalek [Information Sciences]

Ali Maher Talaat Elsayed [Information Sciences]

Hany Mohamed Ali Badr [Information Sciences]

Abdelrahman Abdelhalim El-Aref Hussien [Information Sciences]

Under Supervision of

Dr. Shaimaa Haridy
[Lecturer],
Information Sciences Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

T.A. Radwa Mostafa
[Assistant lecturer],
Information Sciences Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

June 2024

Acknowledgements

All praise and thanks are due to Allah, the Almighty, who has granted us the ability to complete this work. We humbly hope that this effort is accepted in His sight.

We extend our heartfelt gratitude to our parents and family for their unwavering support and encouragement throughout the years of study. Their constant guidance and motivation have been instrumental in our academic achievements.

We would also like to express our sincere appreciation to our esteemed supervisors, Prof. Dr. Hala Mushir, Dr. Shaimaa Haridy, for their invaluable guidance, patience, and expertise throughout the thesis writing process.

Lastly, we extend our thanks to our friends and all those who have offered us support and encouragement along the way. We are deeply grateful for their unwavering faith in us, which has been a source of great motivation and inspiration.

Abstract

In today's digital age, social media platforms have become a prevalent medium for individuals to share their opinions, emotions, and experiences. Despite the surge in user-generated content, effective tools and resources for sentiment analysis in the Arabic language remain insufficient.

This paper addresses this gap by presenting a novel approach to Arabic sentiment analysis through the development of a web application for text emotion classification.

The proposed methodology employs an **ensemble** of deep learning models, including Bidirectional Long Short-Term Memory (**Bi-LSTM**), Bidirectional Gated Recurrent Unit (**Bi-GRU**), and the **MARBERTv2** transformer model, combined using a Random Forest stacking technique. The system's performance is evaluated on the **Emotone_ar** dataset, providing a robust benchmark for emotion detection tasks. Experimental results demonstrate that the ensemble model outperforms individual models, achieving an accuracy of 90%, a recall of 90%, and an F1 score of 90%. The integration of **MARBERTv2**, a pre-trained language model specifically designed for Arabic, shows superior performance compared to other models tailored for the Arabic language.

This work concludes that the proposed ensemble model not only advances the field of Arabic sentiment analysis but also offers an effective tool for real-time emotion detection in social media texts, addressing a critical need in natural language processing for Arabic.

في العصر الرقمي الحالي أصبحت منصات التواصل الاجتماعي وسيلة شائعة للأفراد لمشاركة آرائهم وعواطفهم وتجاربهم. وعلى الرغم من زيادة المحتوى الذي ينشئه المستخدمون، لا تزال الأدوات والموارد الفعالة لتحليل ردود الفعل والآراء والمشاعر باللغة العربية غير كافية.

يعالج هذا البحث هذه الفجوة من خلال تقديم نهج جديد لتحليل المشاعر باللغة العربية عن طريق تطوير تطبيق ويب لتصنيف النصوص بناءً على ما تدل عليه هذه النصوص من مشاعر.

تتضمن المنهجية المقترحة تجميع نماذج التعلم العميق، وهذا يتضمن نموذج الذاكرة طويلة وقصيرة الأمد ثنائية الاتجاه ونموذج وحدة البوابة التكرارية ثنائية الاتجاه ونموذج التحويل، يتم دمج الثلاث نماذج باستخدام تقنية

Random Forrest Stacking

تم تدريب النموذج وتقييم أداء النظام باستخدام مجموعة بيانات

Emotone_AR

تظهر النتائج أن النموذج المجمع النهائي يتفوق على النماذج الفردية بنسبة دقة وصلت إلى 90% وأيضا ساهم الدمج مع نموذج التحويل المدرب مسبقاً بشكل كبير في تحقيق هذه النسبة وذلك لأنه مصمم خصيصاً للغة العربية وأيضا يتفوق على أغلب النماذج الموجودة حالياً والتي تستهدف اللغة العربية.

يخلص هذا العمل إلى أن نموذج التجميع المقترح لا يقدم تقدمًا في مجال تحليل المشاعر باللغة العربية فحسب، بل يوفر أيضًا أداة فعّالة لاكتشاف العواطف في نصوص وسائل التواصل الاجتماعي في الوقت الفعلي، مما يلبي حاجة ماسة في معالجة اللغة الطبيعيةالعربية على اختلاف وتنوع لهجاتها.

Table of Contents

Acknowledgements.. ii

Abstract.. iii

List of Figures.. vii

List of Tables.. viii

List of Abbreviations.. ix

Chapter 1: Introduction.. 10

1.1 Problem Definition.. 10

1.2 Motivation.. 11

1.3 Objectives.. 12

1.4 Methodology 13

1.5 Time plan.. 14

1.6 Thesis Outline 15

Chapter 2: Literature Review 17

Chapter 3: System Architecture and Methods 20

3.1 System Architecture 20

3.2 Description of methods and procedures used 22

Chapter 4: System Implementation and Results 28

4.1 Dataset 29

4.2 Description of Software Tools Used 32

4.3 Setup Configuration (hardware)..35

4.4 Experimental and Results 36

Chapter 5: Run the Application..47

Chapter 6: Conclusion and Future Work..52

6.1 Conclusion..52

6.2 Drawbacks and Future Studies..53

6.3 Future Work……………………………………………………..55
References………………………………………………………….……57

List of Figures

Figure 1.5: Time Plane ….…………………………………………………………15

Figure 3.1: System Architecture ….……………………………...…………....20

Figure 4.1: Distribution of Tweets Across Dataset Column 26

Figure 4.2: Dataset Sample……………………………………………………28

Figure 4.3: Dataset Sample 2 …………………………………………………28

Figure 4.4: BI-GRU Model Architecture ………………..……………………35

Figure 4.5: BI-GRU Model Classification Report ……..………………………35

Figure 4.6: BI-GRU Model Architecture ………………..……………………36

Figure 4.7: BI-LSTM Model Architecture ……………..…..……………………37

Figure 4.8: MARBERTv2 Classification Report ……………………………...43

Figure 4.9: Random Forest Classification Report ……………………………..45

Figure 5.1: Login Page ……………………………………………………...48

Figure 5.2: Signup Page ………………………………………………………48

Figure 5.3: Home Page ……………………………………………………...50

Figure 5.4: Analysis Page …………………………………………………...51

Figure 5.5: History Page ………………………………………………………51

List of Tables

Table 2.1: Table of Literature Review ……………………………………………...18

Table 4.1: Number of Tweets for Emotone_ar After Cleaning ...…………...……27

Table 4.1: Comparison of Our Model to Related Work ...…………………...……38

List of Abbreviations

ANN: Artificial Neural Network

BERT: Bidirectional Encoder Representations from Transformers

BI_GRU: Bidirectional Gated Recurrent Unit

BI_LSTM: Bidirectional Long Short-Term Memory

CNN: Convolutional Neural Network

CLS: Classification

DF: Deep Feature-based model

DL: Deep Learning

GRU: Gated Recurrent Unit

HEF: Human-Engineered Feature-based model

LSTM: Long Short-Term Memory

MARBERTv2: Morphologically Aware Bidirectional Encoder Representations from Transformers version 2

ML: Machine Learning

NLTK: Natural Language Toolkit

NN: Neural Network

RF: Random Forest

SEP: Separation

Chapter One

Introduction

1.1) Problem Definition:

In today's world, social media platforms are flooded with comments and tweets expressing people's thoughts and feelings about different things like products, events, or issues. But going through all these comments manually to understand what people are saying is hard work.

It takes a lot of time, and it's easy to make mistakes. Plus, there are just so many comments being made every day that it's almost impossible for anyone to keep up. This makes it tough for businesses to know what their customers really think about their products, or for analysts to study trends in the market, or for anyone who's just curious about what people are saying about something important happening in the world.

And when things are changing quickly, like during a big news event or a viral trend, trying to keep track of all the comments in real-time is even harder.

So, the big problem we're trying to solve is how to make it easier and faster to understand what people are saying in all these comments and tweets. We want to create a tool that can automatically read through them, figure out how people are feeling, and then show that information in a simple way, like with charts or graphs. This way, businesses can learn more about what their customers want, and everyone else can get a better idea of what's going on in the world around them.
Overall, our project aims to solve the problem of spending too much time and effort on manually reading comments and tweets, by creating a tool that does it automatically and shows the results in an easy-to-understand way.

1.2) Motivation:

1. **Facilitating Data-Driven Decision Making in Real-Time**: Our project addresses the pressing need for businesses, analysts, and individuals to gain actionable insights from the vast amount of user-generated content on social media platforms. By automating the process of sentiment analysis and visualization, we empower stakeholders to make informed decisions in real-time. Businesses can leverage these insights to understand customer feedback, adapt marketing strategies, and enhance brand perception. Analysts gain valuable market intelligence to identify trends and opportunities, while individuals gain deeper insights into societal dynamics and public opinion on various topics. By providing timely and accurate sentiment analysis, our project facilitates data-driven decision-making, fostering agility and responsiveness in an increasingly dynamic digital landscape.

2. **Streamlining Market Research and Customer Feedback Management**: Manual review processes for analyzing comments and tweets pose significant challenges for businesses seeking to conduct market research and manage customer feedback effectively. Our project streamlines these processes by automating sentiment analysis and visualization, enabling businesses to extract valuable insights efficiently. By classifying emotions expressed in user-generated content and presenting them through intuitive charts and graphs, our solution simplifies the interpretation of data, allowing businesses to identify patterns, trends, and sentiment shifts quickly. This streamlined approach enhances the efficiency of market research efforts and enables businesses to respond promptly to customer feedback, ultimately improving brand perception and customer satisfaction.

3. **Leveraging Twitter as a Rich Source of Public Opinion**: Twitter serves as a prolific platform for individuals to express their opinions, thoughts, and sentiments on a wide range of topics, including products, events, and

societal issues. However, manually reviewing the vast volume of tweets generated daily is impractical and time-consuming. Our project recognizes the significance of tapping into this wealth of user-generated content to gain insights into public sentiment and opinion. By extracting tweets from Twitter and automating the process of sentiment analysis and visualization, we unlock a valuable source of real-time data that can inform decision-making across various domains. Whether it's businesses seeking to understand customer perceptions, analysts conducting market research, or government officials monitoring public sentiment, our solution offers a streamlined approach to extracting actionable insights from Twitter's dynamic landscape.

1.3) Aims & Objectives:

1. **Development of an Emotion Detection Model for Tweet Analysis**: The primary objective is to create an emotion detection model capable of analysing tweets to classify the expressed emotions accurately. This model will be integrated into a user-friendly web application interface, enabling users to interact seamlessly with the sentiment analysis functionality. By leveraging machine learning algorithms, natural language processing techniques, and sentiment analysis frameworks, the aim is to automate the extraction and classification of emotions from tweets. This objective aligns with the goal of providing stakeholders, including businesses, analysts, and individuals, with valuable insights into public sentiment and opinion on various topics, products, and events.

2. **Implementation of Web Scraping Feature for Twitter Data Extraction**: Another key objective is to develop a web scraping feature using Selenium WebDriver to extract tweets from Twitter's dynamic platform. This feature will enable the automated retrieval of tweet data based on specified search criteria, ensuring the availability of relevant data for analysis. By leveraging web scraping techniques, the aim is to streamline the data collection process and facilitate the integration of real-time Twitter data into the sentiment analysis workflow. This objective supports the overarching goal of providing

stakeholders with access to up-to-date and comprehensive datasets for sentiment analysis and decision-making purposes.

3. **Creation of a User-Friendly Web Application Interface**: The overarching objective is to design and implement a user-friendly web application interface that facilitates seamless interaction with the emotion detection model and Twitter data extraction feature. The web application will provide users with intuitive controls and visualization tools to explore and analyze tweet sentiment in real-time. Additionally, the interface will offer functionalities for customizing search parameters, visualizing sentiment trends, and exporting analysis results for further exploration. By prioritizing usability and accessibility, the aim is to empower users, including businesses, analysts, and individuals, to gain actionable insights from Twitter data with ease and efficiency. This objective underscores the commitment to democratizing access to sentiment analysis tools and fostering informed decision-making across diverse domains

1.4) Methodologies:

1. **Emotion Detection Model for Tweet Analysis**: Our project's objective is to develop an emotion detection model specifically tailored for analyzing tweets in Arabic. Leveraging a combination of GRU (Gated Recurrent Unit), LSTM (Long Short-Term Memory), and MARBERT (Multilingual Arabic RoBERTa) models, we aim to create an ensemble model capable of accurately classifying the emotions expressed in Arabic tweets. By training this ensemble model on a diverse dataset, including the **Emotone_ar** dataset comprising unlabeled tweets with multiple dialects, we seek to achieve a high level of accuracy in identifying and categorizing emotions such as happiness, sadness, anger, and others. This objective facilitates the automation of sentiment analysis in Arabic tweets, empowering users to gain valuable insights into public opinion and sentiment trends on various topics, products, and events

2. **Web Scraping Functionality for Data Acquisition**: To facilitate the retrieval of relevant tweet data from Twitter's dynamic platform, we developed a web scraping functionality using Selenium WebDriver. This functionality enables

automated data extraction based on user-defined search criteria, ensuring the availability of up-to-date and contextually relevant tweet data for analysis. By leveraging Selenium WebDriver's capabilities, we implemented a robust and efficient web scraping mechanism that navigates Twitter's interface, retrieves tweet content, and stores it for subsequent analysis. This methodology streamlines the data acquisition process, enabling seamless integration of real-time Twitter data into our sentiment analysis workflow. Additionally, our approach ensures compliance with Twitter's terms of service and data usage policies, thereby mitigating potential legal and ethical concerns associated with web scraping activities. Through this methodology, we empower stakeholders to access timely and comprehensive tweet data, facilitating informed decision-making and analysis.

1.5) Time Plan:

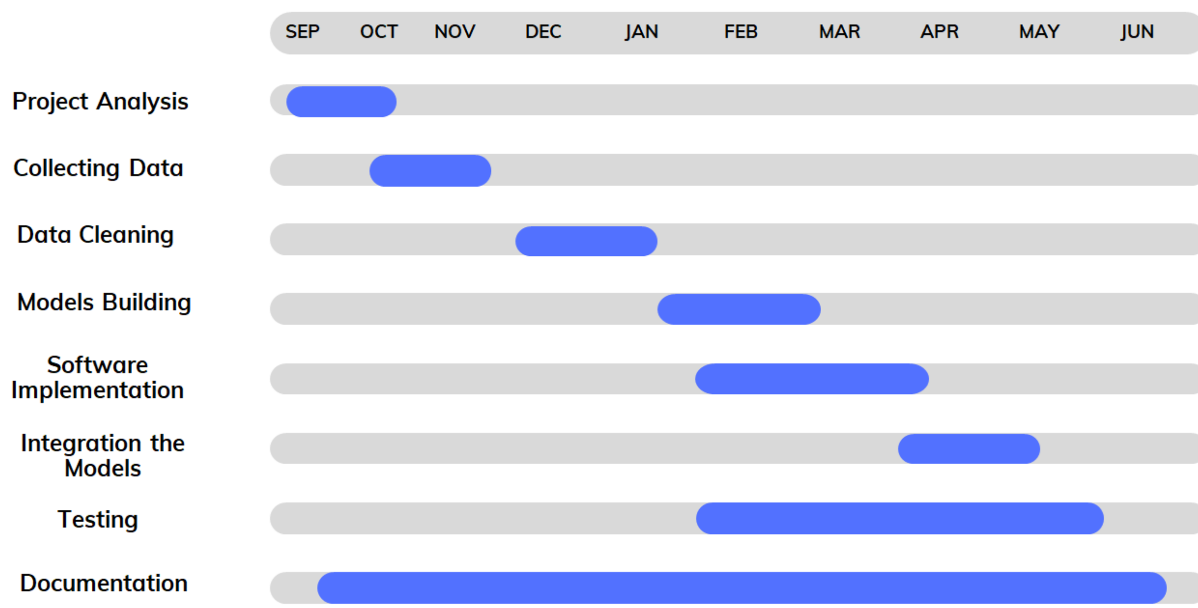


Figure 1.1: Time Plan

1.6) Thesis Outline:

1. Introduction:

- Introduction to the project, highlighting the importance of emotion detection in analyzing tweets and citizen feedback.
- Overview of the aims, objectives, and methodology employed in the project.
- Brief overview of the time plan and expected outcomes.

2. Literature Review:

- Review of existing literature on emotion detection models and sentiment analysis in social media data.
- Exploration of relevant research findings and methodologies in the context of Arabic text classification.
- Identification of gaps in the literature and justification for the current study.

3. System Architecture and Methods:

- Description of the system architecture, outlining how data flows through the application.
- Explanation of the methods used for emotion detection in tweets and sentiment analysis.
- Discussion on the design principles and considerations in developing the system.

4. System Implementation and Results:

- Detailed account of the system implementation process, including tools and technologies utilized.
- Presentation of results obtained from the implementation, emphasizing important patterns and trends.

- Analysis of the system's performance and effectiveness in achieving its objectives.

5. Application Demonstration:

- Showcase of the web application's user interface and features.
- Demonstration of how users interact with the application, with screenshots or video demonstrations.
- Discussion on user feedback and usability testing results

6. Conclusion:

- Summary of the main findings and conclusions drawn from the project.
- Discussion of practical implications and potential applications of the research findings.

7. Future Work:

- Suggestions for future research directions and enhancements to the system.
- Identification of areas for further investigation, such as expanding the dataset or improving classification accuracy.

Chapter Two

Literature Review

The task of sentiment analysis has evolved significantly over the years, starting with basic heuristic-based and lexicon-based approaches then advancing to include machine learning models utilizing TF-IDF representations. More recently, deep learning models and transformers have become state-of-the-art in this field. Despite these advancements, Arabic sentiment analysis has received considerably less attention compared to its English counterpart.

Due to the underrepresentation of Arabic text sentiment analysis research, our review includes a broad range of sentiment analysis works, encompassing the common multilabel sentiment classification. This is pertinent since the core objective in all these works is text emotion classification. To provide a thorough overview of the current research landscape, we have compiled a table summarizing key related works, highlighting their achievements through metrics such as accuracy, F1-score and more.

Reference	Year	Dataset	Methodology	Performance Metrics
An Ensemble Deep Learning Approach for Emotion Detection in Arabic Tweets [1]	2022	SemEval 2018 task 1- Ec-Ar	MARBERT, BI-LSTM, BI-GRU	Accuracy: **0.540** Macro F1 Score: **0.701** Precision: **0.634** Recall: **0.550** Micro F1 Score: **0.527**
Combining Context-aware Embeddings and an Attentional Deep Learning Model for Arabic Affect Analysis on Twitter [3]	2021		AraBERT word embeddings, attention-based LSTM and BI-LSTM	Accuracy: 0.538
Hybrid Feature Model for Emotion Recognition in Arabic Text [2]	2020		HEF + DF Hybrid of human-engineered feature-based model +deep feature-based (DF) model	Micro F1: **0.631** Macro F1: **0.502** Jaccard Acc: **0.512**
Textual Emotions [5]	2023	Emotone _AR	Bi-GRU	Accuracy: **0.72** F1-score: **0.71**
Improved Emotion Detection Framework for Arabic Text using Transformer Models [6]	2023		MARBERT	Accuracy: **0.81** F1-score: **0.71**

Text Based Emotion Recognition in Arabic Text [4]	2019		CNN	F1-score: 0.70 Accuracy: 0.70
			BI-GRU	Accuracy: 0.73 F1-score: 0.74

Table 2.1: Table of Literature Review

Chapter Three

System Architecture and Methods

Introduction:

This chapter provides an in-depth exploration of the system architecture and methodology employed in the context of a Graduation project. The primary objective of this chapter is to establish a strong foundation by outlining the technical framework and approach adopted throughout the project. By elucidating the system architecture and methodologies, readers will gain a comprehensive understanding of the project's structure, components, and the rationale behind the chosen methods.

3.1) System Architecture:

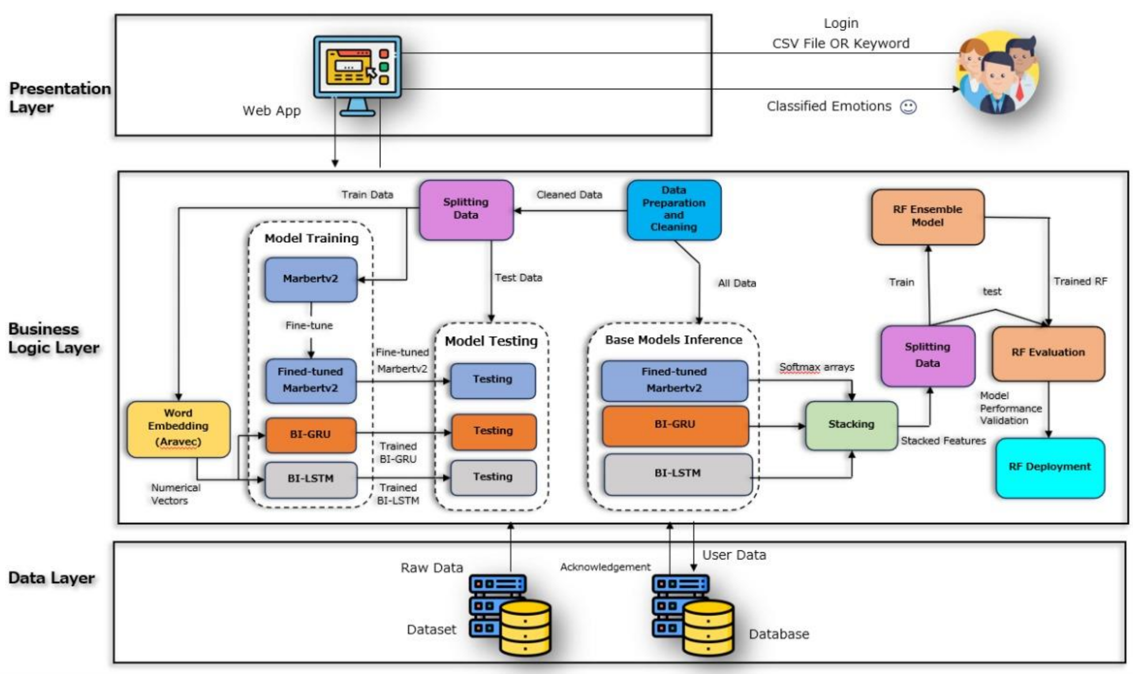


Figure 3.1: System Architecture

Three-Tier System Architecture:

The Application's system architecture is divided into three layers:

1. **Presentation Layer:**

   developed using Flask, the web application interface allows users to log in and classify emotions in text by uploading CSV files or inputting keywords. Users can **upload CSV files** containing tweets or reviews for comprehensive sentiment analysis. Alternatively, they can **search Twitter** by entering specific keywords to retrieve the latest 50 tweets containing those keywords. The results are displayed on an **interactive dashboard** with detailed charts, graphs, and tables, providing insightful analysis of sentiments and trends in the processed text.

2. **Business Logic Layer (Application Layer):**

   The Business Logic Layer, also known as the application layer, serves as the foundation of our platform, handling essential operations on the collected data. It coordinates preprocessing tasks and strategic operations to optimize system efficiency, facilitating seamless data management and advanced analytics capabilities. This layer is responsible for applying business rules, handling data transformations, and coordinating communication between the data access layer and the user interface, thereby enhancing the overall functionality and reliability of the application.

3. **Data Layer:**

   In our application, we utilize a SQL database to store user information and maintain the history of the uploaded files. This database architecture ensures efficient and scalable data management, accommodating the dynamic needs of user interactions. This layer acts as the backbone of the application, providing a reliable foundation for the business logic and presentation layers to operate on.

3.2) Description of methods and procedures used:

The Graduation project employed several methodologies to achieve its objectives, which are outlined below:

3.2.1) Pre-Processing:

We worked on Pre-processing the Arabic reviews dataset. We applied several steps to clean and normalize the text data. Here's the revised description:

1. Removing Stop Words: Common stop words such as: "هم" ,"اللتي" ,"اللذي" ,"من" ,"و", and "هو" were removed from the text. These words have limited semantic value and were excluded from further analysis.

2. Removing punctuation marks such as: "()-[]{};:'",<>./@#$%^&*؛،_~" were eliminated from the text. These symbols do not contribute significantly to the analysis and were removed.

3. Removing English Words and Numbers: Any English words and numerical characters present in the text were disregarded. This step ensured that only Arabic language content was considered for analysis.

4. Normalizing Text: The text was further normalized using the following sub-steps:
   - Removing التشكيل و التطويل: Diacritical marks (tashkeel) and elongation symbols were removed from the text. These symbols were stripped to simplify the text for analysis.
   - Replacing [إأآا] with [ا]: Different forms of the Arabic letter "alif" ([إأآا]) were replaced with the standard form "ا". This normalization step aimed to unify the representation of the same letter in different forms.

5. Removing Extra Spaces and Links: Additional spaces and links, such as "https://drive.google.com/drive/u/0/folders/1ISME6K", were removed from the text. These elements were irrelevant to the analysis.

6. Removing Emails: All email addresses present in the text were removed. This step ensured that personal and irrelevant information, such as "example@example.com", was excluded from the analysis.

7. Removing Mentions: Mentions of usernames, often found in social media texts and denoted with "@" (e.g., "@username"), were eliminated. This helped in focusing on the content of the text rather than specific user references.

8. Removing Diacritics: Diacritical marks (tashkeel) were removed from the text. These marks, which include symbols such as "َ", "ً", "ُ", "ٌ", "ِ", "ٍ", "ْ", and "ّ", were stripped to simplify the text for analysis.

9. Removing New Lines: New line characters (e.g., "\n"," ") were eliminated from the text. This step ensured that the text was treated as a continuous sequence of words, facilitating more effective analysis.

10. Removing Repeating Characters: Sequences of repeating characters were reduced to a single character. For example, "ممتاااز" was normalized to "ممتاز". This helped in standardizing the text and reducing noise.

11. Replacing Emojis: Emojis in the text were replaced with corresponding textual descriptions or removed. This step ensured that non-textual elements, such as "😊" or "❤️", were either represented in a textual format or excluded from the analysis.

3.2.2) **Model's Preparation and Training:**

The model preparation and training phase is a critical component of our sentiment analysis application. This phase involves several steps to ensure that our models are well-trained, optimized, and capable of accurately detecting emotions in Arabic text.
To facilitate the training of deep learning models, we performed the following steps in the dataset preparation phase after the preprocessing step:

For Bi-LSTM and Bi-GRU Models

1. **Word Embedding:** we used AraVec[13] word embeddings specifically the Twitter-based, Twitter-CBOW N-gram model which has Vec-size 300 dimensions, 66,900,000 documents and vocab size of 1,467,715.

   Word embedding is a technique used in natural language processing (NLP) to represent words or phrases in a numerical format. It is a way of transforming textual data into a numerical representation that can be understood and processed by machine learning algorithms.

   Word embedding overcomes these limitations by representing words as dense, low-dimensional real-valued vectors. The key idea behind word embedding is that words with similar meanings or that appear in similar contexts should have similar numerical representations.

   The resulting word embeddings are dense vectors that encode semantic information about the words. Words with similar meanings or that are used in similar contexts will have similar vector representations, allowing the model to capture similarities and relationships between words.

2. **Text vectorization**: It involves converting textual data into numerical vectors that can be processed by neural networks. One commonly used approach for text vectorization in deep learning is the use of tokenization and word embedding techniques.

Tokenization and Input Formatting for MARBERTv2 Model

For the MARBERTv2 model, specific preprocessing steps were performed to prepare the text data for input:

1. **Tokenization and Special Tokens**: The text is broken down into smaller units called tokens. For example:

   - **Original Text**: "الاوليمبياد الجايه هكون لسه ف الكليه .."
   - **Tokenized**: ['الاوليم', '##بياد', 'الجايه', 'هكون', 'لسه', 'ف', 'الكليه', '.', '.']

2. **Special Tokens**:

   - **[CLS] Token**: Added at the beginning of each sentence. This special token stands for "classification" and is used by the model to aggregate information from the entire sentence. The model uses the representation of this token to make predictions.

   - **[SEP] Token**: Added at the end of each sentence. This special token stands for "separator" and is used to mark the end of a sentence. It helps the model understand where the sentence ends, which is particularly useful when dealing with multiple sentences.

   For example, after adding the special tokens:

   - **Original Text**: "الاوليمبياد الجايه هكون لسه ف الكليه .."
   - **Formatted with Special Tokens**:

     '[CLS] الاوليمبياد الجايه هكون لسه ف الكليه .. [SEP]'

3. **Padding and Truncating**: To ensure all sentences have the same length, sentences that are too short are padded with special [PAD] tokens, and sentences that are too long are truncated. This ensures the input data fits within the model's expected input size.

4. **Attention Masks**: These are used to differentiate between real tokens and padding tokens. An attention mask is an array where real tokens are marked with 1s, and padding tokens are marked with 0s. This tells the model which parts of the input should be attended to, and which parts should be ignored.

By performing these steps, we ensure that our text data is correctly formatted for the MARBERTv2 model, allowing it to process the input effectively and make accurate predictions.

Model Training and Evaluation

The training process involves initializing the models, optimizing hyperparameters, and evaluating performance:

1. **Initialization**: Setting up initial parameters and hyperparameters for each model.
2. **Training**: Training the models on the training set using optimization techniques such as Adam or SGD to minimize the loss function.
3. **Validation**: Periodically evaluating the models on the validation set to tune hyperparameters and select the best-performing model.
4. **Testing**: Evaluating the final model ensemble on the test set to determine accuracy and generalization capability, using metrics such as accuracy, precision, recall, and F1-score.

Ensemble Random Forest Stacking Model:

1. **Model Inference**:
   - Perform inference using the trained BI-LSTM, BI-GRU, and fine-tuned **Marabertv2** models on the entire dataset to generate SoftMax probability arrays.

2. **Stacking**
   - **SoftMax Aggregation:** Aggregate the SoftMax outputs from BI-LSTM, BI-GRU, and fine-tuned **Marabertv2** models to create a combined feature set for ensemble modelling.

3. **Data Splitting:** Split the aggregated data into training and testing datasets for training and evaluating the Random Forest ensemble model.

4. **Model Training:** Train a Random Forest ensemble model using the training dataset from the stacking step to combine predictions and improve overall performance.

5. **Model Testing:** Test the Random Forest ensemble model on the testing dataset to assess its accuracy and F1-score on unseen data.

Conclusion:

In conclusion, this chapter provided a detailed explanation of the system architecture implemented, also a detailed explanation of the preprocessing techniques applied to the dataset, including stop word removal, punctuation removal, normalization, and word replacement. It also emphasized the importance of word embedding and text vectorization for effectively feeding deep learning models with Arabic tweets, aiming to achieve optimal accuracy.

Chapter Four

System Implementation and Results

Introduction to this Chapter:

In this chapter, we will delve into the implementation of our system and present the results obtained. The chapter will cover the following key aspects:

1. **Dataset:** We will provide an overview of the dataset utilized for training and evaluating our proposed models. The dataset plays a crucial role in achieving accurate and reliable emotion detection.
2. **Description of Software Tools Used:** We will discuss the various software tools employed in the development of our application, highlighting their significance in achieving our project goals. These tools have been carefully selected to ensure efficiency and effectiveness.
3. **Setup Configuration:** We will explore the necessary setup configuration required for the smooth functioning of the application. This includes hardware specifications, software dependencies, and any other relevant configurations.
4. **Experimental and Results:** In this section, we will present the experimental procedures conducted to evaluate the performance of our system. We will discuss all experiments undertaken to achieve these results, providing a detailed account of the methodologies employed, including any variations and adjustments made throughout the evaluation process. The chapter will showcase the results obtained, providing insights into the effectiveness and potential of our application.

Overall, this chapter will provide a comprehensive overview of the system implementation process and present the exciting results achieved.

4.1 Dataset:

Our dataset, named 'Emotone_ar,' is a collection of Arabic tweets annotated with eight emotions: sadness, anger, joy, surprise, love, sympathy, fear, and none. The dataset was created by researchers at Nile University in Egypt and was first published in 2017. It contains 10,065 tweets collected from Twitter using a set of keywords related to each emotion. The tweets were then manually annotated by three native Arabic speakers.

Additionally, it's important to note that this dataset is multi-dialect, encompassing various Arabic dialects to ensure its applicability across different regions and linguistic variations. Furthermore, emotone_ar is a balanced dataset, with approximately equal numbers of tweets for each emotion. This balance enhances its suitability for training and evaluation purposes, ensuring that models developed using this dataset are well-rounded and effective across different emotional categories.

Distribution of Tweets Across Dataset Columns:

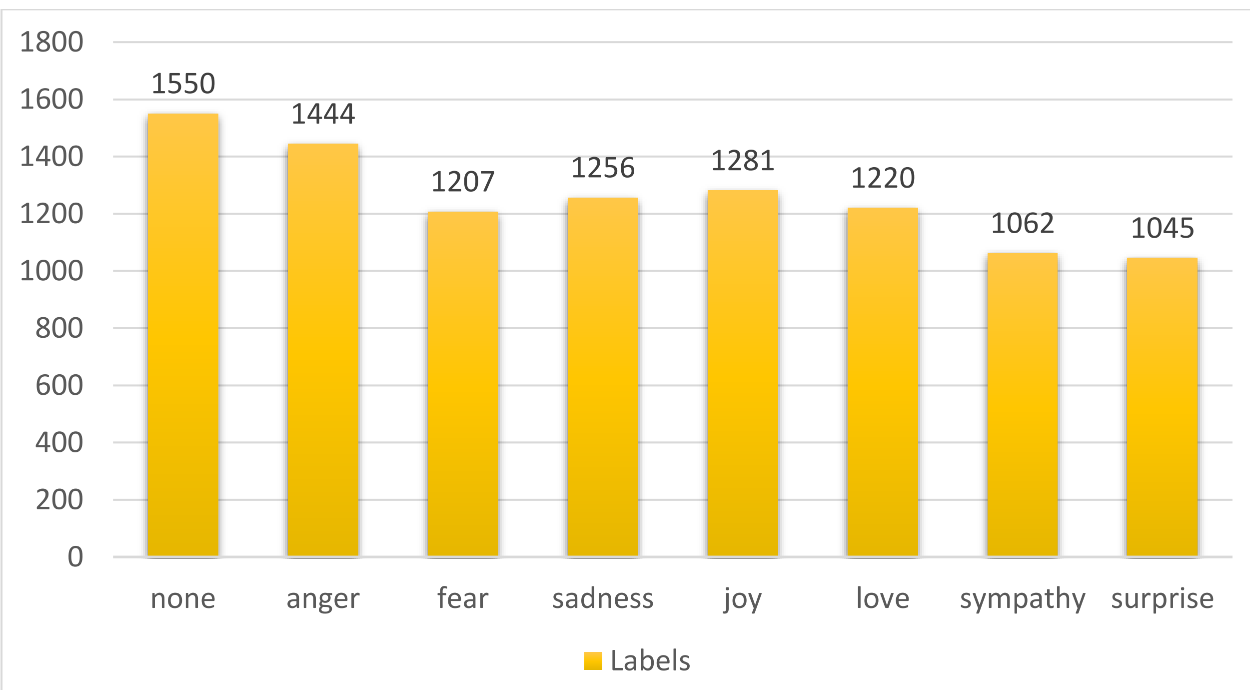


Figure 4.1: Distribution of Tweets Across Dataset Columns

Dataset Column Distribution After Preprocessing and Cleaning:

Table 4.1: Number of Tweets for Emotone_ar After Cleaning

Emotion	Count
None	1539
Anger	1440
Fear	1204
Sadness	1254
Joy	1280
Love	1213
Sympathy	1046
Surprise	1044
Total	10,020

During the preprocessing and cleaning phase of our dataset, we removed null and duplicated rows in the dataset to improve models' accuracy.

Data Samples Before and After Cleaning:

	ORIGINAL_TWEET	PROCESSED_TWEET
0	.. الاوليمبياد الجايه هكون لسه ف الكليه	الاوليمبياد الجايه هكون لسه الكليه
1	...عجز الموازنه وصل ل93.7 % من الناتج المحلي يعني	... عجز الموازنه وصل ل الناتج المحلي يعني لسه اقل
2	xD كتنا نيله ف حظنا الهباب	كتنا نيله حظنا الهباب
3	...جميعنا نريد تحقيق اهدافنا لكن تونس تالقت في حر	جميعنا نريد تحقيق اهدافنا تونس تالقت حراسه المرمي
4	... الاوليمبياد نظامها مختلف .. ومواعيد المونديال	...الاوليمبياد نظامها مختلف ومواعيد المونديال مكا
...	...	...
10015	...يلا يا جماعه حفله عمرو دياب خلصت نريح شو :2222	... يلا جماعه حفله عمرو دياب خلصت نريح شويه ونبدا
10016	❤️اييييه دااا 😲😲 اوزيييل :Mohamed5	ايه دا اوزيل قلب
10017	😃 عملتلها ريتويت بمناسبه ساره بتاعه الاوليمبياد	... عملتلها ريتويت بمناسبه ساره بتاعه الاوليمبياد
10018	وعليك قبلنا يانجم النجوم ياعندليب الحب والاحساس	قبلنا يانجم النجوم ياعندليب الحب والاحساس
10019	...يطلع ننهم كل شي سيء ووضيع كل خساسه الع AlHamad	...يطلع نهم شي سيء وضيع خساسه العالم تجمعت الايرا

Figure 4.2: Dataset Samples

	TWEET	anger	fear	joy	love	none	sadness	surprise	sympathy
0	الاوليمبياد الجايه هكون لسه الكليه	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
1	... عجز الموازنه وصل ل الناتج المحلي يعني لسه اقل	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	كتنا نيله حظنا الهباب	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
3	جميعنا نريد تحقيق اهدافنا تونس تالقت حراسه المرمي	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
4	...الاوليمبياد نظامها مختلف ومواعيد المونديال مكا	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
10015	... يلا جماعه حفله عمرو دياب خلصت نريح شويه ونبدا	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
10016	ايه دا اوزيل قلب	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
10017	... عملتلها ريتويت بمناسبه ساره بتاعه الاوليمبياد	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
10018	قبلنا يانجم النجوم ياعندليب الحب والاحساس	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
10019	...يطلع نهم شي سيء وضيع خساسه العالم تجمعت الايرا	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 4.3: Dataset Samples 2

4.2 Description of Software Tools Used:

1. **Seaborn:** An open-source Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

2. **NumPy:** A fundamental package for scientific computing with Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions. We used it to save our models.

3. **Pandas:** A powerful data manipulation and analysis library, offering data structures and operations for working with structured data sets such as Data Frames.

4. **Sklearn Library for Dataset Management, Splitting, One-Hot Encoding, and Classification Report:**
   - The Sklearn library, also known as Scikit-learn, is a popular Python library used for machine learning tasks. It provides essential tools for dataset management, including data splitting into training and testing subsets. Additionally, Sklearn offers efficient techniques for one-hot encoding categorical variables and generating classification reports to evaluate the performance of machine learning and deep learning models.

5. **PyTorch Library for Deep Learning Model Development:**
   - Developed by Facebook’s AI Research lab, PyTorch is a flexible and user-friendly open-source library for building deep neural networks. With dynamic computational graphs and automatic differentiation, PyTorch is ideal for research and development in deep learning projects. Its comprehensive tools and APIs allow for easy experimentation with different network architectures and training algorithms, making it suitable for various tasks such as image classification, object detection, and natural language processing.

PyTorch integrates seamlessly with other libraries like NumPy and TensorFlow, enabling efficient data manipulation and interoperability.

6. **NLTK for Natural Language Processing:**
    - NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources, along with a suite of text processing libraries for tasks such as tokenization, stemming, tagging, parsing, and semantic reasoning.
    - NLTK's extensive collection of tools and resources makes it an invaluable asset for researchers, educators, and practitioners in the field of natural language processing (NLP), enabling them to perform a wide range of text analysis tasks efficiently and effectively.
7. **TensorFlow for Deep Learning Model Development:**
    - TensorFlow is a widely used deep learning framework that provides a comprehensive platform for building and training neural networks. Its flexible architecture allows for easy experimentation with different network architectures and training algorithms.
    - TensorFlow offers extensive support for deep learning tasks such as natural language processing. Additionally, TensorFlow provides high-level APIs like Keras, which simplify the process of building and training neural networks, making it accessible to developers of all skill levels.
8. **Gensim Library for Topic Modeling and Document Similarity Analysis:**
    - It provides efficient implementations of various algorithms for analyzing large text corpora, including Word2Vec.
    - Gensim's algorithms enable users to extract meaningful topics from text data, identify similarities between documents, and perform other text analysis tasks. With its intuitive interface and high-performance implementations.
9. **Colab (Google Colaboratory):** A free, cloud-based platform provided by Google for running Python code, particularly well-suited for deep learning tasks with its access to GPUs and TPUs.

10. **Kaggle:** An online platform for data science competitions, datasets, and notebooks, provided us with GPU resources, accelerating the training of our deep learning and machine learning models. This expedited experimentation and enhanced the performance and effectiveness of our models.
11. **Hugging Face:** A platform offering state-of-the-art natural language processing models and libraries, including transformers for pre-trained language models. We utilized it for importing our dataset and the Marbert model.

12. **Flask for Web Development:**
    - Flask is a lightweight Python web framework known for its simplicity and ease of use in building web applications and APIs. It provides developers with the necessary tools and libraries for efficiently creating scalable and robust web services.
    - Built on the WSGI specification and Jinja2 templating engine, Flask offers a solid foundation for web development. Its modular design allows for easy customization, with developers able to add or remove components as needed.
    - Flask's extensive ecosystem of extensions enhances its capabilities, offering additional functionalities such as database integration, authentication, and RESTful API support. It’s simple API design makes it accessible to developers of all skill levels, promoting clean and maintainable code.
    - For graduation projects involving web development, Flask provides a reliable and efficient solution. Its simplicity, flexibility, and wide range of extensions make it a preferred choice for delivering high-quality web services.

4.3 Setup:

PC Specifications:

- Processor: Intel(R) Core (TM) i5-2540M CPU @ 2.60GHz 2.60 GHz
- Installed Memory (RAM): 16.0 GB (15.9 GB Usable)
- System Type: 64-Bit Operating System, x64 Based Processor
- Windows Specifications:
  - Edition: Windows 10 Pro
  - Version: 22H2

Minimum System Requirements:

- Processor: A dual-core processor with a clock speed of at least 2.0 GHz or higher.
- RAM: A minimum of 4 GB of RAM for smooth operation.
- Operating System: Compatible with Windows 10 (64-bit), macOS Mojave (10.14) or later, or Ubuntu 18.04 LTS.

Please note that these are minimum requirements, and higher specifications may be beneficial for improved performance, especially when handling large volumes of requests.

Tools Software Requirements:

Git:

- Operating System: Windows, macOS, Linux
- Processor: 1.5 GHz or faster processor
- RAM: Minimum 512 MB (1 GB recommended)
- Disk Space: Minimum 50 MB available disk space
- Git Client: Install the Git client specific to your operating system (e.g., Git for Windows, Git for macOS)

VS Code:

- Operating System: Windows, macOS, Linux
- Processor: 64-bit processor
- RAM: Minimum 512 MB (1 GB recommended)
- Disk Space: Minimum 200 MB available disk space
- Memory: Minimum 2 GB of RAM.

4.4 Experimental and Results:

Introduction:

The Experimental and Results section presents the detailed outcomes of the models implemented for Arabic Sentiment Analysis. The models evaluated include BI-LSTM, BI-GRU, MARBERTv2, and Ensemble Random Forest Stacking. Additionally, this section covers the selection and preprocessing of datasets used in the experiments

Dataset selection and preprocessing

1. Data selection:

Finding an Arabic dataset for text emotion classification that is manually annotated and large enough for deep learning models is challenging. We identified two suitable datasets for our project: ‘SemEval-2018 Task 1’ and ‘Emotone_ar’.

1.1 SemEval-2018 Task 1

The ‘SemEval-2018 Task 1’ dataset is a collection of tweets annotated for emotion intensity and valence. It is a multilingual dataset, with tweets in English, Arabic, and Spanish. The dataset consists of three subtasks, one of which is:

- **Emotion Classification:** Classifying the tweet as one or more of eight emotions.

The Arabic portion of the dataset contains over 4,000 tweets. These tweets are annotated for emotion intensity, valence, and emotion classification. The dataset is freely available for use.

1.2 Emotone_ar

As mentioned in **section 4.1**, the ‘Emotone_ar’ dataset is a collection of Arabic tweets. It is a large and diverse dataset, with over 10,000 tweets manually annotated with eight emotions.

We have chosen to start our work with the ‘Emotone_ar’ dataset due to its larger number of tweets and manual annotations. We plan to work with the ‘SemEval-2018 Task 1’ dataset in the future.

2. Data preprocessing:

We ensured that the data contained no null values, redundancies, or duplicates, and that it was consistent throughout.

Our preprocessing steps included cleaning the text by removing emails, URLs, mentions, hashtags, punctuation, and other unnecessary elements. Additionally, we normalized the Arabic text and removed diacritics, repeating characters, newlines, stop words, emojis, English characters, and digits.

Models Description and Performance Metrics:

1. BI-GRU:

The BI-GRU is a variant of the GRU (Gated Recurrent Unit) network, designed to efficiently capture sequential dependencies in text data. Unlike traditional RNNs, the BI-GRU processes the input data in both forward and backward directions, enabling it to understand the context more comprehensively. This bidirectional approach is particularly advantageous for sentiment analysis, where the meaning of a word can be influenced by both preceding and succeeding words.

Performance Metrics

Trial 1: Utilized different stemming techniques, including ISRI Stemmer, Snowball Stemmer, and Arabic Light Stemmer. The best accuracy achieved was **67%** using Snowball Stemmer.
Trial 2: Removed the stemming step, resulting in an improved accuracy of **70%**.
Trail 3: Experimented with various model architectures. The best architecture achieved an accuracy of **72%** with an F1-Score of **71%**.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 300, 300)          9702300

 bidirectional_4 (Bidirecti  (None, 300, 256)          330240
 onal)

 dropout_6 (Dropout)         (None, 300, 256)          0

 bidirectional_5 (Bidirecti  (None, 256)               296448
 onal)

 dropout_7 (Dropout)         (None, 256)               0

 dense_7 (Dense)             (None, 8)                 2056

=================================================================
Total params: 10331044 (39.41 MB)
Trainable params: 10331044 (39.41 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 4.4: BI-GRU Model Architecture

Classification report for the model

```
Bi-GRU Model Classification Report:
              precision    recall  f1-score   support

       Anger       0.68      0.81      0.74       136
        fear       0.98      0.85      0.91       112
         joy       0.64      0.50      0.56       117
        love       0.73      0.83      0.78       125
        None       0.67      0.89      0.77       171
     Sadness       0.54      0.52      0.53       123
    surprise       0.71      0.42      0.53       107
    sympathy       0.92      0.85      0.88       111

    accuracy                           0.72      1002
   macro avg       0.74      0.71      0.71      1002
weighted avg       0.73      0.72      0.71      1002
```

Figure 4.5: BI-GRU Model Classification Report

2. **BI-LSTM:**
   The BI-LSTM is a type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data. Unlike traditional RNNs, which process data in one direction, BI-LSTM processes data in both forward and backward directions, enabling it to understand the context from both past and future states. This bidirectional approach helps in better capturing the semantic meaning of each word in the sentence.

Performance Metrics

Trials: Experimented with various model architectures. The best architecture achieved an accuracy of **72%** with an F1-Score of **71%**.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 300, 300)          9702300

 bidirectional_3 (Bidirecti  (None, 300, 600)          1442400
 onal)

 global_max_pooling1d_1 (Gl  (None, 600)               0
 obalMaxPooling1D)

 dense_4 (Dense)             (None, 128)               76928

 dropout_4 (Dropout)         (None, 128)               0

 dense_5 (Dense)             (None, 32)                4128

 dropout_5 (Dropout)         (None, 32)                0

 dense_6 (Dense)             (None, 8)                 264

=================================================================
Total params: 11226020 (42.82 MB)
Trainable params: 11226020 (42.82 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 4.6: BI-LSTM Model Architecture

Classification report for the model

Bi-LSTM Model Classification Report:

	precision	recall	f1-score	support
Anger	0.68	0.80	0.73	136
fear	0.91	0.87	0.89	112
joy	0.75	0.46	0.57	117
love	0.73	0.76	0.75	125
None	0.68	0.89	0.77	171
Sadness	0.58	0.52	0.55	123
surprise	0.59	0.53	0.56	107
sympathy	0.92	0.85	0.88	111
accuracy			0.72	1002
macro avg	0.73	0.71	0.71	1002
weighted avg	0.73	0.72	0.72	1002

Figure 4.7: BI-LSTM Classification Report

3. MARBETv2:

MARBERTv2 is a large-scale pre-training masked language model focused on both Dialectal Arabic and Modern Standard Arabic (MSA). It was trained on a vast dataset to better understand the nuances and variations in Arabic text.

Model Details

- **Dataset:** 1B Arabic tweets from a dataset of 6B Arabic tweets.
- **Text Data:** 128GB of text, totalling 15.6B tokens.
- **Architecture:** Using the same architecture as BERT-base.

Required Formatting for Fine-Tuning

1. Tokenization
2. Adding Special Tokens (SEP & CLS)
3. Truncating/Padding sentence to max length
4. Mapping tokens to their IDs
5. Creating attention mask
6. Returning a dictionary of outputs

7. Converting training and validation data to PyTorch tensors

Training

Hyperparameters and Configuration:

1. Batch Size:
   - Recommended: 16 or 32
2. Learning Rate (AdamW Optimizer):
   - Options: 5e-5, 3e-5, or 2e-5
   - Default in code: 5e-5
3. Number of Epochs:
   - Recommended: 2, 3, or 4
   - Default in code: 2
4. Epsilon (Adam Optimizer):
   - Default: 1e-8
5. Gradient Clipping:
   - Norm of the gradients clipped to 1.0 to prevent "exploding gradients"
6. Learning Rate Scheduler:
   - Linear Scheduler with Warmup
   - Number of warmup steps: 0 (default)
   - Total number of training steps: calculated based on the number of epochs and the size of the training dataset

Training Loop:

1. Initialization:
   - Instantiating the MARBERTv2 classifier model, optimizer (AdamW), and learning rate scheduler.
   - Moving the model to GPU if available.

2. Training:

   - For each epoch, iterate through the training data.
   - For each batch, perform the following steps:

     1. Loading batch to GPU.
     2. Zero out any previously calculated gradients.
     3. Performing a forward pass to compute logits.
     4. Computing loss using cross-entropy loss function.
     5. Performing a backward pass to calculate gradients.
     6. Clipping gradients and update parameters using the optimizer.
     7. Updating learning rate using the scheduler.
     8. Printing training progress every 20 batches, including batch loss and elapsed time.

3. Evaluation (Optional):

   - If evaluation is enabled, evaluate the model on the validation set after each epoch.
   - Compute validation loss and accuracy.
   - Print evaluation results including validation loss and accuracy.

4. Logging:

   - Print the header of the result table for each epoch.
   - Print average training loss, validation loss, and validation accuracy at the end of each epoch.

Training Execution:

1. Initialization:

   - Call the *initialize_model* function to initialize the model, optimizer, and scheduler with the specified number of epochs.
   - Default number of epochs: 2

2. Training and Evaluation:
    - Call the train function to train the model using the training data.
    - Enable evaluation to monitor performance on the validation set after each epoch.
    - Default number of epochs: 2
    - Enable evaluation: evaluation=True

These details provide a complete understanding of the model fine-tuning process for MARBERTv2, including the hyperparameters, training loop, and evaluation steps.

Classification report for the model

	precision	recall	f1-score	support
none	0.79	0.80	0.79	159
anger	0.83	0.82	0.83	148
joy	0.67	0.70	0.69	122
sadness	0.72	0.70	0.71	105
love	0.87	0.82	0.84	122
sympathy	0.88	0.95	0.91	111
surprise	0.72	0.69	0.70	116
fear	0.95	0.94	0.95	124
accuracy			0.81	1007
macro avg	0.80	0.80	0.80	1007
weighted avg	0.81	0.81	0.80	1007

```
Micro F1-score: 0.8052990127700117
Macro F1-score: 0.8034725215741207
```

Figure 4.8: MARBERTv2 Classification Report

4. Ensemble Random Forest Stacking:

The Ensemble Random Forest Stacking method combines the predictions of multiple robust base models to enhance overall performance. This approach is particularly well-suited when the base models are strong learners, as is the case in our project. We utilized the 3 complex models we developed (Bi-LSTM, Bi-GRU and MARBERTv2) as base models. The outputs of these models were used as input features for a Random Forest classifier, which served as the meta-learner, effectively capturing and integrating the diverse strengths of each base model.

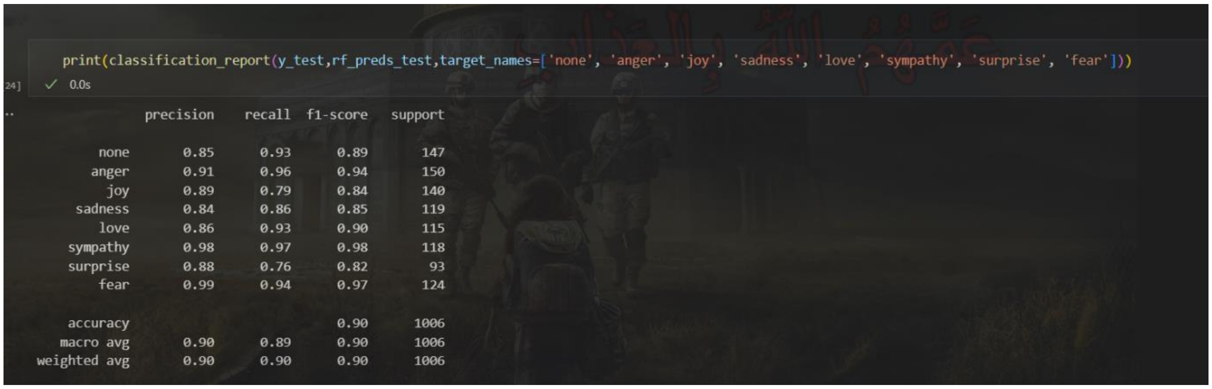
Base Models Preparation

To effectively implement the Ensemble Random Forest Stacking method, it is crucial to prepare and structure the outputs of our base models. In this phase, we performed inference using our base models on the entire dataset. The prediction outputs (SoftMax arrays) from each model were stacked to form a comprehensive feature set. These stacked features were then split into a training set and a test set, ready to be fed into the Random Forest classifier.

Performance Metrics

Trial 1: We started with the default hyperparameters for the Random Forest classifier. The performance metrics were relatively low, with accuracy at 70% and F1-score at 72%.
Trial 2: We performed a Grid Search to optimize the hyperparameters, specifically n_estimators and max_depth. This resulted in improved performance metrics, surpassing the strongest base model. The accuracy increased to 85% and the F1-score to 83%.
Trial 3: Through multiple rounds of experimentation and hyperparameter tuning, we achieved an accuracy of 90% and an F1-score of 90%. **These results significantly exceed the current state-of-the-art in Arabic sentiment analysis**. The detailed classification report is shown in Figure 4.8.

Classification report for the model



```
print(classification_report(y_test,rf_preds_test,target_names=['none', 'anger', 'joy', 'sadness', 'love', 'sympathy', 'surprise', 'fear']))

              precision    recall  f1-score   support

        none       0.85      0.93      0.89       147
       anger       0.91      0.96      0.94       150
         joy       0.89      0.79      0.84       140
     sadness       0.84      0.86      0.85       119
        love       0.86      0.93      0.90       115
    sympathy       0.98      0.97      0.98       118
    surprise       0.88      0.76      0.82        93
        fear       0.99      0.94      0.97       124

    accuracy                           0.90      1006
   macro avg       0.90      0.89      0.90      1006
weighted avg       0.90      0.90      0.90      1006
```

Figure 4.9: Random Forest Classification Report

<u>Comparing our model to the model in related works:</u>

Our model achieves an accuracy of 90%, surpassing the state-of-the-art in Arabic Sentiment Analysis.

<table>
<tr><th>Name</th><th>Date</th><th>Dataset</th><th>Model</th><th>Accuracy</th><th>F1-score</th></tr>
<tr><td rowspan="4">Our Proposed Project</td><td rowspan="4">2024</td><td rowspan="7">Emotone-AR[4]</td><td>BI-LSTM</td><td>0.72</td><td>0.71</td></tr>
<tr><td>BI-GRU</td><td>0.72</td><td>0.71</td></tr>
<tr><td>MARBERT</td><td>0.81</td><td>0.80</td></tr>
<tr><td>Ensemble RF</td><td>0.90</td><td>0.90</td></tr>
<tr><td>Textual Emotions [5]</td><td>2023</td><td>BI-GRU</td><td>0.73</td><td>0.74</td></tr>
<tr><td>Improved Emotion Detection Framework for Arabic Text using Transformer Models [6]</td><td>2023</td><td>Arabic-Bert-base model</td><td>0.74</td><td>0.74</td></tr>
<tr><td>Text Based Emotion Recognition in Arabic text [4]</td><td>2019</td><td>CNN</td><td>0.70</td><td>0.70</td></tr>
</table>

Table 4.1: Comparison of Our Model to Related Work

Chapter Five

Run the Application

Introduction:

This chapter guides users and administrators on running the application, covering searching on Twitter about the recent tweets and classifying it based on the emotions they express, and viewing functionalities. It offers comprehensive instructions for users to effectively search on Twitter or just enter a phrase to know the emotion behind it or you can even upload a .csv file that includes some tweets and download it after classification and also making some data analysis techniques and that because we offer some visualization on the tweets after classification to make the analysis process easier on the user. By following these guidelines, users can efficiently engage with the application, fostering effective Arabic emotion detection.

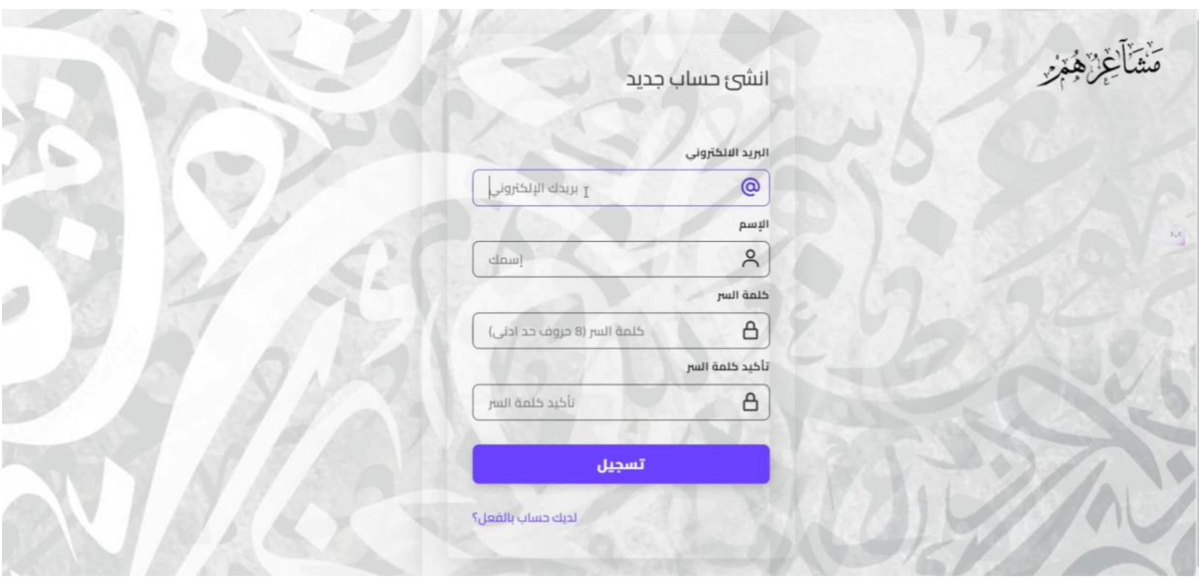
Login page:

As shown in this figure: textbox1 and textbox2: are for the user to enter his username and password. And, if he is a new user he can sign in by clicking on "ليس لديك حساب؟" that is shown below. And the login button if the user enters the right username and password will direct you to the home page. And if he clicked on "ليس لديك حساب؟" it will direct you to the signup page.

(Figure 5.1: Login Page)

Sign-up page:

On the signup page, the first textbox is for the email of the user, the second textbox is for the name of the user, the third textbox is for the password of the user and the last textbox is for the confirmation of the password. Then you can click on the "تسجيل" button, so the data gets saved, and the account gets created.



(Figure 5.2: Sign-up)

Home page:

On the home page the multiple things you can do:

1st) The rectangle on the right hand represents the textbox you can enter the keyword in to search with it on the Twitter, this is done using three options:

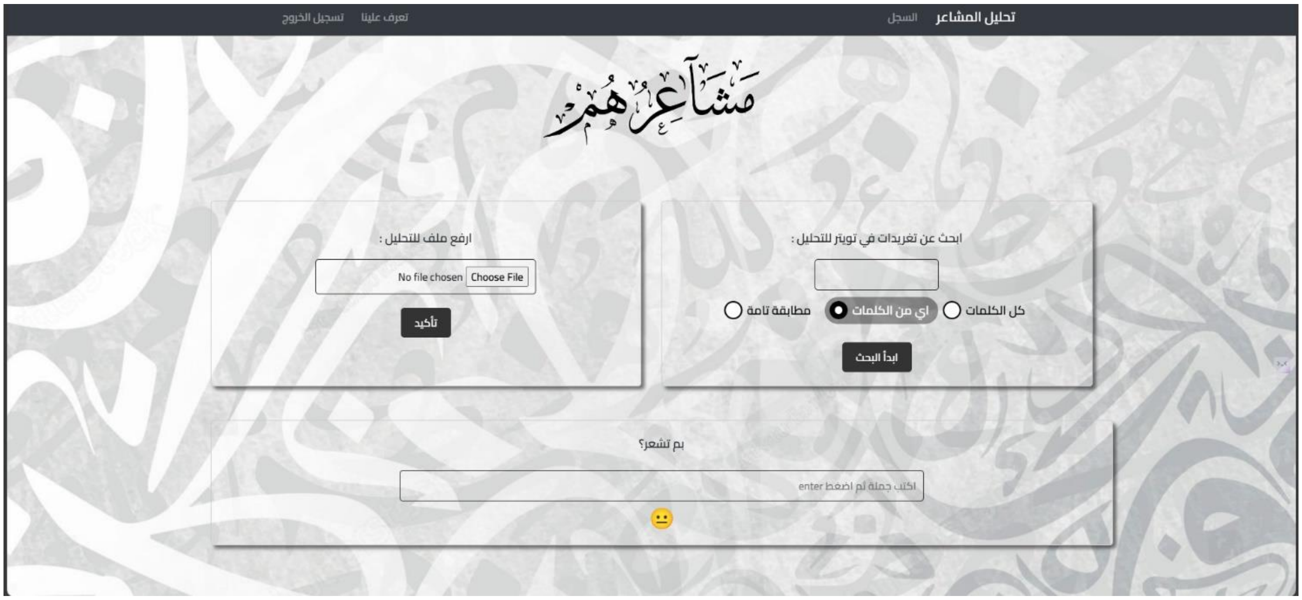
1) You can enter a keyword that consists of more than one word and search using any of these words. This option you can choose by clicking on "أي من الكلمات" from the radio-box.
2) You can enter any keyword and search about it as you entered it even if it is more than one word, so it is an “Exact match” option. This option you can choose by clicking on "مطابقة تامة" button.
3) You can enter a keyword that consists of more than one word and search using all the words that are in the keyword individually. This option you can choose by clicking on "كل الكلمات" button.

After entering the keyword and choosing one of the previous options the application will redirect you to the analysis page with the results of the emotion detection process.

2nd) The rectangle on the left hand represents the upload .csv file section, you can upload .csv file consists of some not classified tweets on the application and click on "تأكيد" then it will redirect you to the analysis page with the results of the emotion detection process.

3rd) The rectangle below makes the user able to insert any Arabic phrase and click “Enter” to get the emotion that represents the phrase the user entered.

4th) Navigation bar contains the “Log-out” button, “Contact Us” link and “History Page” link.



(Figure 5.3: Home Page)

Analysis page:

This page showing the results of the emotion detection process through two active visualization techniques:

1) **Pie Chart**
2) **Bar Chart**

   This chart shows the number of tweets in each emotion and that makes the user know which the more/fewer emotion is appearing in the returned tweets or in the .csv file, and the user can gain sights and make analysis with such a data.

3) **Percentage Table**

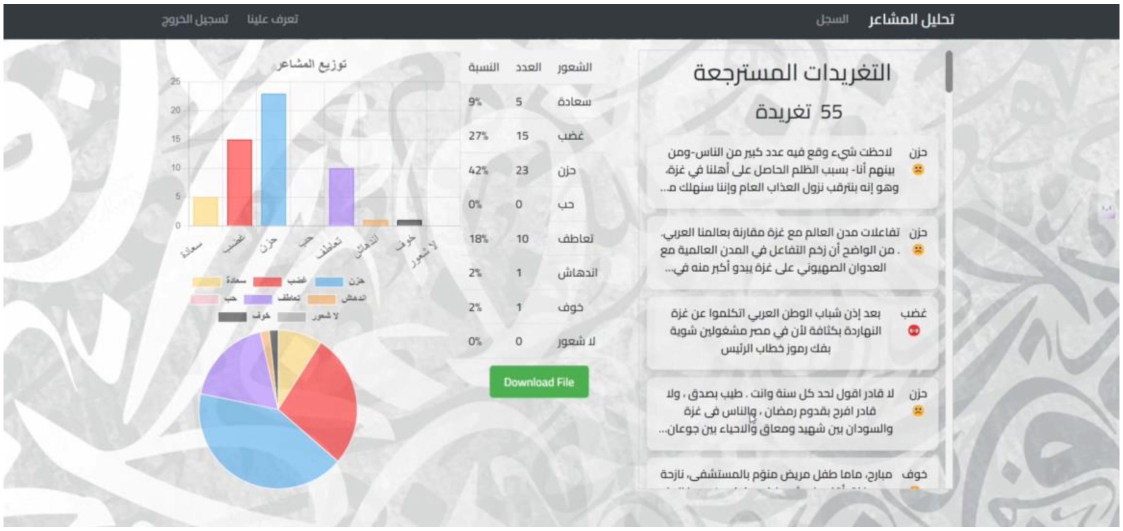
   This table shows the number of tweets in each emotion and the percentage of this number based on the total numbers of the tweets.

4) **Download File Button**

   Button for downloading the classified tweets in .csv file format.

5) **Table of the returned tweets**

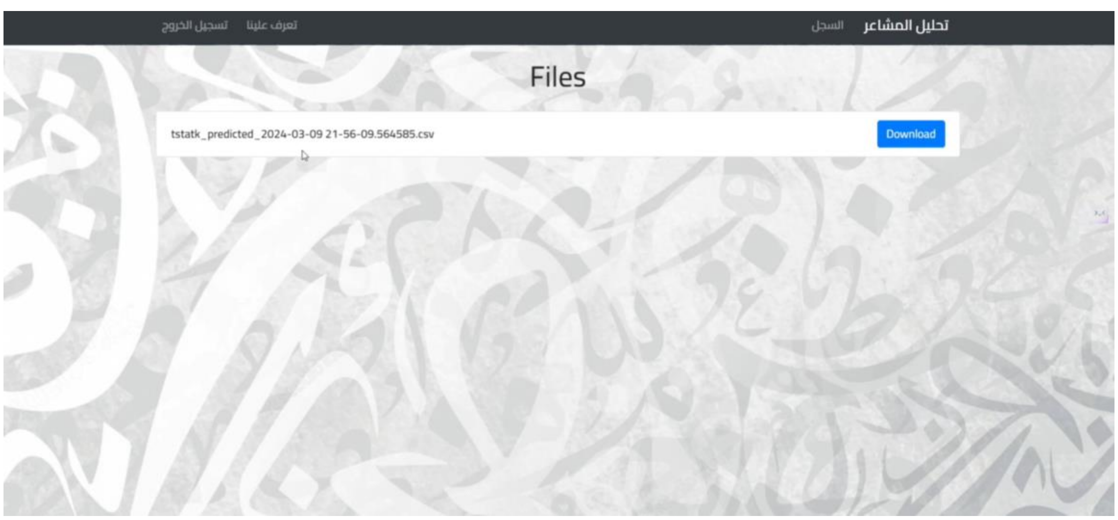
<u>In the case of searching with Twitter</u>. This section shows all the returned tweets and each tweet with its emotion that represents.



(Figure 5.4: Analysis Page)

<u>History page:</u>

As shown in figure 5.5 we can see the file created in the history page and you can download it again, also you can see the date and the time this file created through the file’s name.



(Figure 5.5: History Page)

Chapter Six

Conclusion and Future Work

6.1) Conclusion:

Our web application for Arabic sentiment analysis demonstrates significant advancements in the field of natural language processing (NLP), particularly tailored for Arabic text. Utilizing sophisticated models such as Bidirectional Gated Recurrent Units (Bi-GRU), Bidirectional Long Short-Term Memory (Bi-LSTM), MARBERT, and Random Forest in a stacking ensemble approach, we achieved a high accuracy rate of 90%. This robust performance underscores the effectiveness of our methodology in accurately detecting and classifying a wide range of emotions, including love, anger, fear, and more.

The application is designed with user convenience and versatility in mind, offering two primary functionalities. Users can upload Excel sheets containing textual data to receive a comprehensive analysis, complete with a detailed dashboard and an option to download the enriched dataset with emotion classifications. Alternatively, users can search for specific keywords to retrieve and analyze the latest 50 tweets from Twitter, providing real-time insights into public sentiment. The resulting dashboards present a clear and intuitive visualization of the sentiment distribution, making it easy to interpret and utilize the data.

Our Arabic sentiment analysis web application not only showcases technological innovation but also provides practical, real-world benefits across multiple domains, enhancing the understanding and application of sentiment analysis in the Arabic-speaking world.

Importance of application:

- **Aligns with Egypt's Vision 2030:** Supports digital transformation initiatives.
- **User-Friendly Interface:** Simplifies sentiment analysis for data scientist and normal users.
- **Public Sentiment Insights:** Analyzes recent tweets and textual data for informed decision-making.
- **Data-Driven Decisions:** Provides detailed dashboards and data classifications.
- **Enhances Customer Feedback Management:** Identifies areas of satisfaction and concern.
- **Real-Time Social Media Analysis:** Keeps users updated on trends and sentiments.
- **Promotes Arabic NLP Research:** Utilizes advanced models like BI-GRU, BI-LSTM, and MARBERTv2.
- **Streamlined Workflows:** Offers Excel sheet analysis and downloadable results.
- **Business Consumer Insights:** Helps businesses understand consumer emotions.
- **Comprehensive Data Visualization:** Easy interpretation of sentiment analysis results.

6.2) Drawbacks and Future Studies:

Despite the high accuracy and robust performance of our Arabic sentiment analysis web application, there are several limitations to our approach that need to be addressed in future studies:

- **Dataset Limitation**:
  - Our model is trained on a single dataset of Arabic tweets. While the results are promising, the performance of our model may not generalize well to other datasets of Arabic tweets or different forms of Arabic text, such as news articles, blogs, or comments. This limitation highlights

the need for extensive testing on diverse datasets to ensure broader applicability and reliability.

- **Context Ignorance**:
    - The current model does not consider the context of the tweets. Emotions expressed in tweets can be heavily influenced by the surrounding context, including preceding conversations, cultural nuances, and current events. Ignoring these contextual factors may lead to misclassification of emotions. Future work should focus on integrating contextual understanding into the model, potentially using advanced NLP techniques like context-aware transformers or attention mechanisms.

- **Limited Dataset for Emotion Detection**:
    - The dataset used for training the emotion detection model is limited. A more extensive collection of real-life interaction data is needed to improve the model's performance and generalizability. Gathering a larger and more diverse dataset, including data from various Arabic dialects and different social media platforms, will enhance the model's ability to accurately detect emotions in a wide range of scenarios.

- **Dialect and Language Variations**:
    - Arabic has many dialects that can vary significantly from one region to another. Our model might not perform equally well across all dialects due to variations in vocabulary, syntax, and expressions. Future studies should aim to include a more representative sample of different Arabic dialects to improve the model's robustness.

- **Handling Sarcasm and Irony**:
    - Detecting sarcasm and irony remains a challenging task in sentiment analysis, especially in informal social media texts. Our current model

may struggle with accurately identifying these nuanced expressions. Future research should explore techniques specifically designed to handle sarcasm and irony in Arabic text.

- **Integration with Real-time Systems**:
  - While our application provides real-time sentiment analysis of tweets, integrating this functionality into larger real-time monitoring and decision-making systems could pose challenges. Ensuring scalability, speed, and accuracy in high-traffic environments will be crucial for practical deployment.

6.3) Recommendations and Logical solutions for future works:

To address these drawbacks and enhance the capabilities of our Arabic sentiment analysis application, future studies should focus on the following areas:

- **Diverse Dataset Collection**:
  - Collect and annotate a more extensive and diverse dataset covering various forms of Arabic text, including different dialects and sources beyond Twitter. This will help in creating a more comprehensive training corpus.

- **Contextual Analysis**:
  - Develop methods to incorporate contextual information into the sentiment analysis model. This could involve leveraging context-aware NLP models and incorporating additional metadata from tweets.

- **Advanced NLP Techniques**:
  - Explore the use of advanced NLP techniques, such as transformers and attention mechanisms, to improve the model's ability to understand nuanced language features like sarcasm and irony.

- **Dialect-Specific Models**:
    - Create models tailored to specific Arabic dialects to improve accuracy and relevance across different Arabic-speaking regions.

- **Real-time System Integration**:
    - Work on integrating the sentiment analysis model into real-time monitoring systems, ensuring that it can handle high volumes of data with minimal latency.

- **User Feedback and Iterative Improvement**:
    - Implement mechanisms for user feedback to continuously improve the model. This can involve active learning where the model is periodically retrained on newly annotated data based on user input.

By addressing these limitations and focusing on future improvements, we can further enhance the accuracy, reliability, and applicability of our Arabic sentiment analysis web application, making it a more powerful tool for understanding and analyzing emotions in Arabic text.

REFERENCES:

[1] Alaa Mansy, Sherine Rady, and Tarek Gharib. "An ensemble deep learning approach for emotion detection in arabic tweets." International Journal of Advanced Computer Science and Applications 13.4 (2022).

[2] N. Alswaidan and M. E. B. Menai, "Hybrid Feature Model for Emotion Recognition in Arabic Text," in IEEE Access, vol. 8, pp. 37843-37854, 2020, doi: 10.1109/ACCESS.2020.2975906.

[3] H. Elfaik and E. H. Nfaoui, "Combining Context-Aware Embeddings and an Attentional Deep Learning Model for Arabic Affect Analysis on Twitter," in IEEE Access, vol. 9, pp. 111214-111230, 2021, doi: 10.1109/ACCESS.2021.3102087.

[4] Lababidi, Haya Ahmad Saad, Algendy, Mai Hany, Ahmad, Yasmen Mahmoud, and Ahmad, Yomna Osama. "Text Based Emotion Recognition in Arabic Text." June 2019. Ain Shams University, Faculty of Computer & Information Science, Computer Science Department.

[5] Abd El Baky, Mahmoud Nasser, Borai, Youssef Mohamed, Hilaly, Youssef Mohamed, Kamel, Youssef Mohamed, and Wadie, Ramy Anwar. "Textual Emotions." Documentation submitted for the degree of Bachelor's in Computer and Information Science, June 2023. Ain Shams University, Faculty of Computer and Information Sciences, Information System Department.

[6] Noaman, Hatem. "Improved Emotion Detection Framework for Arabic Text using Transformer Models." Advanced Engineering Technology and Application 12.2 (2023): 1-11.

[7] Alqarni, Arwa, and Atta Rahman. "Arabic Tweets-based Sentiment Analysis to investigate the impact of COVID-19 in KSA: A deep learning approach." Big Data and Cognitive Computing 7.1 (2023): 16.

[8] Al Shamsi, Arwa A., and Sherief Abdallah. "Ensemble Stacking Model for Sentiment Analysis of Emirati and Arabic Dialects". Journal of King Saud University-Computer and Information Sciences (2023): 101691.

[9] Fatima Khalil Aljwari "Emotion Detection in Arabic Text using Machine Learning Method". Article in IJISCS (International Journal of Information System and Computer Science) · December 2022 DOI: 10.56327/ijiscs.v6i3.1322

[10] Elfaik, Hanane, and El Habib Nfaoui. "Deep bidirectional LSTM network learning-based sentiment analysis for Arabic text." Journal of Intelligent Systems 30.1 (2020): 395-412.

[11] Al-Bayati, Abdulhakeem Qusay, Ahmed S. Al-Araji, and Saman Hameed Ameen. "Arabic sentiment analysis (ASA) using deep learning approach." Journal of Engineering 26.6 (2020): 85-93.

[12] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "MARBERT: Deep Bidirectional Transformers for Arabic," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 7088-7105, 2021. Online. DOI: 10.18653/v1/2021.acl-long.551.

[13] A. B. Soliman, K. Eisa, and S. R. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP," in Proceedings of the 3rd International Conference on Arabic Computational Linguistics (ACLing 2017), Dubai, UAE, 2017.

[14] A. Al-Khatib and S. El-Beltagy, "Emotional Tone Detection in Arabic Tweets," in Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2017), Budapest, Hungary, 2017.