NIRE: CLASSIFYING THE EMOTIONAL LANDSCAPE OF TWEETS IN SAUDI TOURISM

ASMA MOHAMMED ALAIDROUS
TAHANI ABDULLAH ALMUTAIRI

SUPERVISED BY: DR. NADA OAMR BAJUNAID

DEPARTMENT OF COMPUTER SCIENCE
KING ABDULAZIZ UNIVERSITY

February, 2026

NIRE: CLASSIFYING THE EMOTIONAL LANDSCAPE OF TWEETS IN SAUDI TOURISM

ASMA MOHAMMED ALAIDROUS
TAHANI ABDULLAH ALMUTAIRI

DEPARTMENT OF COMPUTER SCIENCE
KING ABDULAZIZ UNIVERSITY

February, 2026

NIRE: CLASSIFYING THE EMOTIONAL LANDSCAPE OF TWEETS IN SAUDI TOURISM

ASMA MOHAMMED ALAIDROUS
TAHANI ABDULLAH ALMUTAIRI

THIS REPORT IS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE BACHELOR DEGREE
IN COMPUTER SCIENCE

SUPERVISED BY: DR. NADA OAMR BAJUNAID

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF COMPUTERS INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY

February, 2026

Declaration of Originality

We hereby declare that this project report is based on our original work except for citations and quotations, which had been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at KAU or other institutions.

Student Name	ID	Signature	Date
Asma Mohammed Alaidrous	1914941	AM	February, 2026
Tahani Abdullah Almutairi	1906666	TA	February, 2026

Acknowledgements

First and foremost, we extend our sincerest gratitude to Allah. We sincerely thank our esteemed supervisor, Dr. Nada Bajunaid. Her unwavering support, insightful wisdom, and tireless encouragement have been instrumental in our progress. Our sincere appreciation goes to our committee members, Dr. Mai Fadel, Dr. Arwa Basbrain, Dr. Ohoud Alzamzami, and our coordinator, Prof. Arwa Alaama. Their expert advice, valuable feedback, and continuous guidance have significantly contributed to the successful completion of this project. Our deepest thanks are extended to our families and friends, whose steadfast support and belief in our abilities have been our driving force throughout this journey. Lastly, our experience of working together as a team of two has been enriching. Despite the challenges faced, the bond we developed, the mutual respect, and the collaborative spirit between us played a pivotal role in completing this project. Our shared knowledge and constant encouragement helped us progress, step by step, toward our shared goal. This project has not only culminated in an output but has also left us with invaluable lessons about collaboration, perseverance, and mutual support. We are deeply thankful for the experience.

Abstract

The tourism industry in Saudi Arabia has rapidly expanded in recent years, playing a vital role in the country's economic success. The government and authorities have promoted the country's attractions and events to a broader audience through social media platforms like Twitter. However, with this expansion, a deluge of public opinions and reviews has emerged, making it difficult to manually analyze and respond to them effectively. This situation limits businesses' and authorities' ability to understand their audience's preferences, issues, and suggestions, potentially impacting strategic decision-making and customer satisfaction. Recognizing this problem, our project proposes Nire, a web-based tool that utilizes advanced Natural Language Processing (NLP) techniques and deep learning algorithms to perform Aspect-Based Sentiment Analysis (ABSA) on Saudi tourism-related tweets. By automating the review analysis process, Nire eliminates the labor-intensive manual task, ensuring more accurate and faster results. This tool, particularly focused on entertainment tourism and hospitality tourism, helps capture customer opinions and identify patterns or trends in the data. ABSA in our project consists of three main tasks: aspect term extraction, aspect category classification, and aspect sentiment classification. Each component was implemented as a separate model then integrated into our Nire web-based tool for comprehensive analysis. The aspect term extraction model achieved an F1-score of 73%, the aspect category model yielded an F1-score of 75%, and the aspect sentiment model obtained an F1-score of 94%. With NLP and deep learning, Nire empowers individuals and businesses in the hospitality industry, and soon in the entertainment industry, to monitor and address negative comments or concerns immediately.

المستخلص

توسع قطاع السياحة في المملكة العربية السعودية بشكل سريع في السنوات الأخيرة، وقد لعب دورًا حيويًا في النجاح الاقتصادي للبلاد. تعمل الحكومة والسلطات على الترويج للمناطق الجاذبة والترفيهية في الدولة لجمهور أوسع من خلال مختلف منصات وسائل التواصل الاجتماعي، مثل تويتر. ومع هذا التوسع، ظهر كم هائل من الآراء العامة والتعليقات، مما صعب من مهمة تحليلها والرد عليها بشكل فعال يدويًا. وقد تقيد هذه القدرة على فهم تفضيلات الجمهور ومشاكلهم واقتراحاتهم من قبل الشركات والسلطات مما يؤثر على اتخاذ القرارات الاستراتيجية والاهتمام برضا العملاء. مع التعرف على هذه المشكلة، يقترح مشروعنا نير أداة على الويب تستخدم تقنيات متقدمة لمعالجة اللغة الطبيعية وخوارزميات التعلم العميق لإجراء تحليل المشاعر القائم على الجوانب في التغريدات المتعلقة بالسياحة السعودية. من خلال تشغيل عملية تحليل الرأي، يتخلص نير من العمل اليدوي المكلف، مما يضمن نتائج أكثر دقة وسرعة. تركز هذه الأداة بشكل خاص على سياحة الترفيه وسياحة الضيافة، وتساعد في تحليل آراء العملاء وتحديد الأنماط أو الاتجاهات في البيانات. يتكون تحليل المشاعر القائم على الجوانب في مشروعنا من ثلاث مهام رئيسية: استخراج مصطلح الجانب، تصنيف فئة الجانب، وتصنيف المشاعر حسب الجانب. تم تنفيذ كل مهمة كنموذج منفصل، ثم تم دمجهم في أداة نير للتحليل الشامل. حقق استخراج مصطلح الجانب نسبة تقريبية تبلغ ٧٣%، بينما حقق نموذج تصنيف فئة الجانب نسبة تقريبية تبلغ ٧٥% ونموذج تصنيف المشاعر حسب الجانب نسبة تقريبية تبلغ ٩٤%. بفضل تقنيات معالجة اللغة الطبيعية وخوارزميات التعلم العميق، يمكن لنير مساعدة الأفراد والشركات في قطاع الضيافة، وقريبًا في قطاع الترفيه، في مراقبة ومعالجة التعليقات السلبية أو المخاوف على الفور.

Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

ABSA	Aspect-Based Sentiment Analysis.
ACC	Aspect Category Classification.
AI	Artificial Intelligence.
ANLP	Arabic Natural Language Processing.
ANN	Artificial Neural Network.
API	Application Programming Interface.
ASC	Aspect Sentiment Classification.
ATE	Aspect Term Extraction.
BERT	Bidirectional Encoder Representations from Transformers.
BiGRU	Bidirectional Gated Recurrent Unit.
BIO	Begin-Inside-Outside.
CBOW	Continuous Bag of Word.
CNN	Convolutional Neural Network.
CRF	Conditional Random Field.
CSV	Comma-Separated Value.
DA	Dialectal Arabic.
DL	Deep Learning.
DNN	Deep Neural Network.
ER	Entity Relationship.
GRU	Gated Recurrent Unit.
KNN	K-Nearest Neighbor.
LR	Logistic Regression.
LSTM	Long Short-term Memory.
ML	Machine Learning.
MSA	Modern Standard Arabic.
NB	Naïve Bayes.
NLP	Natural Language Processing.
NLTK	Natural Language Toolkit.
POS	Part of Speech.
RNN	Recurrent Neural Network.
SA	Sentiment Analysis.
SemEval	Semantic Evaluation International Workshop.
SPOS	Stanford Part of Speech.
SVM	Support Vector Machine.
TF-IDF	Term Frequency-Inverse Document Frequency.

CHAPTER 1

INTRODUCTION

Social media platforms significantly impact people's lives today, including in Saudi Arabia. In the tourism sector, the government and authorities have promoted the country's attractions and unique cultural experiences to a broader audience through various social platforms. Twitter is used to carry out these activities through the Ministry of Tourism of Saudi Arabia's official account. This example of using Twitter to promote tourism in Saudi Arabia inspired us to investigate the possibilities of utilizing Twitter to gather and analyze people's opinions to facilitate further growth and satisfaction within the tourism industry.

This chapter introduces the context and motivation for this project and provides the problem, followed by a suggested solution. Additionally, this chapter outlines the project's aim, objectives, scope, and target users. Finally, the project's methodology and plan are presented.

1.1 Context and Motivation

The expansion of the tourism industry in Saudi Arabia has played an essential role in the country's recent economic success. It has also been instrumental in the progress toward reaching the goals of the Kingdom's Vision 2030 plan. This noticeable growth offers the potential for the Kingdom to become one of the world's leading tourism industry countries.

According to (Vision2030, nd) website, the Kingdom organized more than 3,800 tourism events attended by more than eighty million people worldwide. The Ministry of Tourism provided options that suit all segments of society, and different income levels, strengthening social ties by providing opportunities for

families and friends to share their fun times.

Therefore, no surprise that many attendants have left their digital footprints about their overall experience on one of the major social media platforms like Twitter. Many individuals use Twitter to express or share their experiences concerning an accommodation, attraction, or destination. It may be about a positive or a negative experience. Positive tweets can help promote the event or attraction to a larger audience, while negative tweets can damage an organization's reputation and discourage potential attendees.

For that reason, event organizations need to monitor tweets and address any negative comments or concerns to maintain a positive reputation and foster customer loyalty.

1.2 Problem Definition

Twitter is a valuable tool for tourism, allowing travelers to exchange opinions and recommendations when deciding on vacation destinations. Tourism companies and organizations can utilize this platform to better understand travelers' needs and preferences, enabling them to tailor their marketing and offerings accordingly. Conversely, individuals can use Twitter to gather information about different vacation destinations and accommodations, helping them to make more informed decisions. However, manually analyzing tweets can be time-consuming and subjective as it relies on the judgment of the person conducting the analysis. Thus, to reduce the time and effort, Aspect-Based Sentiment Analysis (ABSA) techniques were utilized. These techniques combined Natural Language Processing (NLP) concepts, Deep Learning (DL) algorithms, and text mining methods.

1.3 Suggested Solution

This project proposed a web-based tool for performing ABSA on Saudi tourism-related tweets. The solution aims to gain insight into the opinions and emotions of customers or other stakeholders and identify patterns or trends in the data. The tool utilized DL models trained for this task to classify the sentiment expressed in tweets concerning aspects of Saudi tourism. Different visualizing formats were used to convey the analysis results to users.

The tool allowed users to filter the results by sentiment for a more comprehensive understanding and data analysis. By uploading a file containing a set of tweets, users received an analysis of the sentiment expressed towards specific aspects mentioned in the tweets. The proposed solution is illustrated in Figure 1.1.

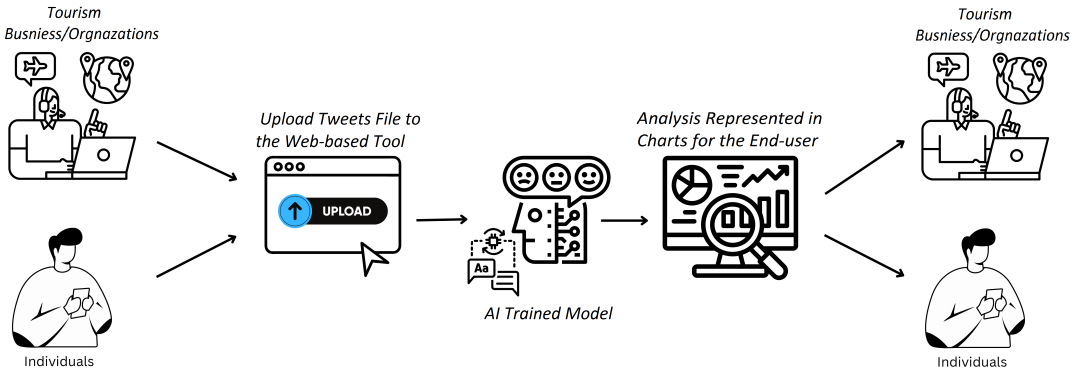

Figure 1.1: The Proposed Solution.

1.4 Project Aim and Objectives

This project aimed to develop a web-based system that analyzes public opinion on aspects of Saudi tourism and hospitality through Twitter. Using ABSA, the tool helps make informed decisions and improve the overall tourism experience in the country with less time and effort. Towards this project's aim, the following objectives were achieved:

1. Conduct a literature review to identify existing methods and approaches for ABSA, particularly in Arabic texts.

2. Collect and pre-process a dataset of tweets related to Saudi entertainment tourism.

3. Build DL models to identify aspects mentioned in each tweet and the sentiment of each one of the aspects.

4. Develop a web-based interface that integrates with the DL models that analyzes the uploaded tweets by the user.

1.5 Target Users

The target users of this project are organizations and businesses in the tourism and hospitality industry in Saudi Arabia. Our tool can be beneficial, especially for start-up businesses in the industry; such companies need an affordable alternative to manual analysis. Furthermore, the project targets individuals seeking to understand how the public feels about tourist destinations or accommodations.

1.6 Project Scope

This project involved using DL algorithms and NLP techniques to analyze sentiments expressed in Arabic tweets regarding specific aspects of the tourism industry in Saudi Arabia, with a particular focus on entertainment and hospitality tourism. The project only considers uploaded tweets by the user, and aims to provide insights into attitudes and opinions related to specific aspects such as services, prices, hotels, location, and facilities.

1.7 Methodology

In this project, we used the iterative waterfall methodology for software development. According to (Kaur and Kumar, 2015), This approach is based on the traditional waterfall model but allows for iterative revisions. The stages included in this methodology are listed below in Figure 1.2.

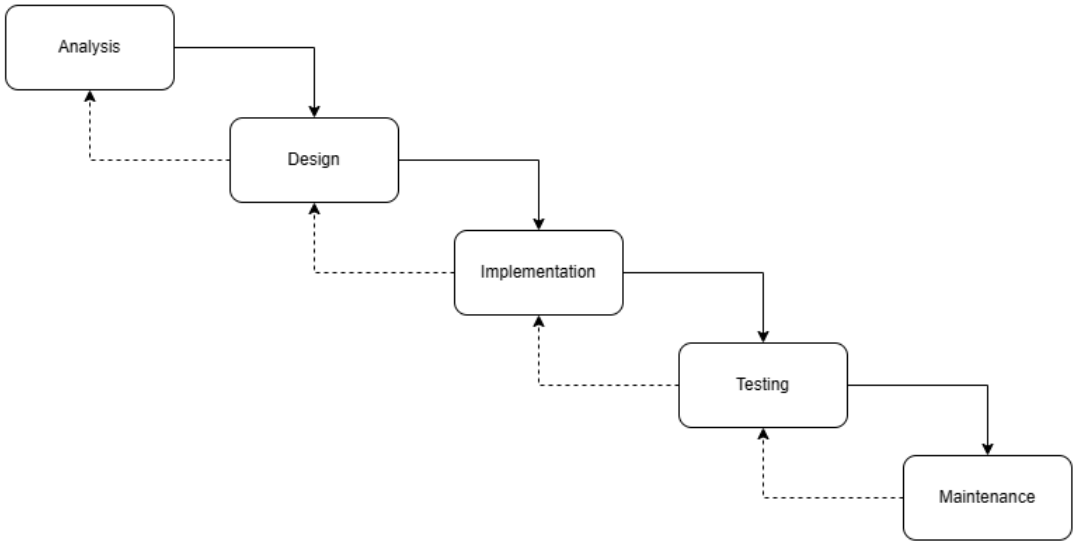

Figure 1.2: Iterative Waterfall Methodology.

This method allows for greater flexibility and adaptability during the development process. The team can make changes and improvements as they work rather than wait until the project's end to address any problems. It can reduce the risk of rework and improve the final product's quality.

While it may not be suitable for highly unpredictable or rapidly changing requirements, the iterative waterfall model is ideal for projects with strict deadlines, such as ours. It provides a clear road map for completing the work within a predetermined time frame. By following this defined process, our team was able to easily track the progress and identify potential issues or delays throughout the project.

1.8 Project Plan

Following the project methodology, the project plan is scheduled for the academic year 2023. As shown in Table 1.1 the milestones of this project have been achieved through each chapter. Chapter 1 introduced the project thoroughly. Chapter 2 covered the literature review. Chapter 3 explained the data collection and requirements. Whereas chapter 4 introduced the system design and analysis. The first three milestones are part of CPCS498 senior project 1, and CPCS499 senior project 2 covered the rest.

Table 1.1: The Project Plan and Milestones.

Semester	**Stages**	**Milestone**	**Tasks**	**Duration**
First Semester	Analysis	Introduction	Problem Definition & Proposed Solution	2 Weeks
			Objectives, Scope, and Target Users	
		Literature Review	Read Scientific Papers in the Project Domain	1 Week
			Analyze the Related Work	1 Week
	Design	System Design and Analysis	System Analysis	6 Weeks
			Requirement Specification	
			Dataset Collection	
			Dataset Annotation	
			Prototype Design	
			Prototype Test	
Second Semester	Implementation	Application Design and Analysis	Dataset Pre-processing	7 Weeks
			Model Implementation	
			Model Improving	
			Model Testing	
		Developing the Website	Building Front-end	
			Building Back-end	
	Testing	Website Integration and Testing	Website and Model Integration	3 Weeks
			Testing	
			Deploy	

1.9 Conclusion

The purpose of this chapter was to provide an overview of our project. We began by discussing the project's context and motivation, the problem definition, and the suggested solution. The aim and objectives were set, then identified the target users and defined the scope of the system. Finally, we defined the timeline of the project. The next chapter provides a literature review of the techniques used in the project.

CHAPTER 2

LITERATURE REVIEW

The integration of Artificial Intelligence (AI) and Natural Language Processing (NLP) has led to significant advancements in the field of human-computer interaction. By utilizing Machine Learning (ML) algorithms to analyze and interpret natural language, it is now possible to build natural and intuitive systems that can communicate and understand humans. NLP techniques enable rapid and accurate analysis of large quantities of customer feedback or social media data, allowing for a wide range of new Sentiment Analysis (SA) applications. These include understanding and tracking public opinion and sentiment on Twitter. In this project, we explored the use of NLP and ML algorithms to analyze Twitter data and gained insights into public sentiment about current advances in the entertainment and hospitality tourism sector in Saudi Arabia. In this chapter, we provide an overview of the main areas of AI relevant to this project's scope. First, a background of relevant AI areas is introduced in multiple sequential sections. Afterward, section 2.7 and section 2.8 summarize the related work. While section 2.9 addresses the software tools. Lastly, section 2.10 concludes this chapter.

2.1 Machine Learning

ML is an AI application that enables systems to acquire knowledge and improve without explicit programming by learning from examples or previous experiences (Alpaydin, 2020). ML can also be described as a collection of techniques that can automatically locate patterns in data and utilize those patterns to forecast future data or help make several decisions (Murphy, 2012). As further explained by the author, machines can be programmed to recognize relationships and

patterns in input data, handle repetitive operations, and aid tasks with enormous computational power. Typically, there are two main categories of ML: supervised and unsupervised learning (Murphy, 2012).

- Supervised learning: In order to predict upcoming events, supervised learning algorithms use labeled data to apply what was learned in the past to new data (Murphy, 2012). The labeled data is known as the training set. The machine develops a model, an algorithmic equation for determining an output using new data called the testing set. After understanding the rules and patterns of the training set, the output is determined based on the rules generated (Theobald, 2021). There are many supervised learning algorithms, such as regression analysis, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and others (Theobald, 2021).

- Unsupervised learning: (Theobald, 2021) states that unsupervised learning does not categorize all variables and data patterns. Instead, unsupervised learning methods must help the system find hidden patterns and generate labels. The system can never guarantee that the output is accurate. As opposed to that, it infers what the result should be. The clustering algorithm is one of the most frequently used unsupervised learning algorithms. Data points with comparable characteristics are grouped using this technique.

- Semi-supervised learning: As explained by (Yang et al., 2022), there is a third type called semi-supervised learning. Over the past ten years, semi-supervised learning has attracted much attention in ML. Semi-supervised learning combines supervised and unsupervised learning using labeled and unlabeled data.

2.2 Deep Learning

As illustrated in Figure 2.1, Deep Learning (DL) is a specialized branch of ML that is designed to handle complex and large datasets that traditional ML algorithms struggle with (Dang et al., 2020). It involves training Artificial Neural Networks (ANNs) to recognize patterns in data through multiple layers of interconnected nodes that perform various stages of information processing. An

activation function is applied to the input data in each layer, enabling the neural network to learn about the non-linear relationships in the data (Dang et al., 2020).

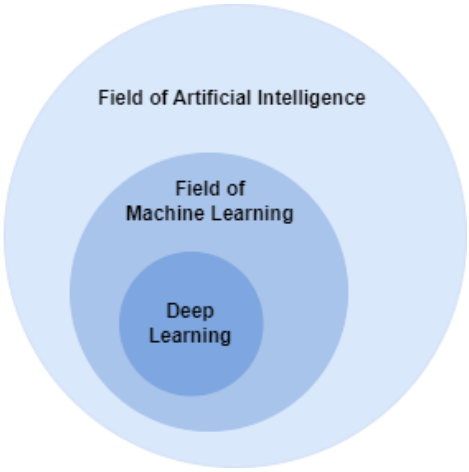

Figure 2.1: DL is a Subset of ML and AI.
Source: (Kayid et al., 2018)

DL networks typically have three main layers: input, hidden, and output. The input layer obtains raw data and sends it to the hidden layer for processing. The hidden layer performs computations on the input data, with the number and size of these layers depending on the complexity of the task (Hu et al., 2021).

For tasks involving sequential data, recurrent layers are often used in the hidden layer to maintain an internal memory of the input data and produce output sequences. For example, in NLP, recurrent layers generate a sequence of words that form a coherent sentence (Hu et al., 2021).

The output layer of a neural network produces the final output, which could be a probability distribution over different classes in a classification task or a set of continuous values in a regression task (Dang et al., 2020). The choice of the loss function depends on the specific task at hand, while the optimizer function updates the model's parameters to minimize the loss and prevent overfitting. When a model is excessively complicated and closely fits the training data, it leads to overfitting, which causes sub optimal performance on new data. However, DL is continually evolving and is expected to overcome the challenges it currently faces, such as the need for large amounts of data and computational resources (Dang et al., 2020).

2.3 Recurrent Neural Networks

ANN with recurrent connections is known as a Recurrent Neural Network (RNN), which can model sequential data such as natural language text or time series (Graves, 2013). As explained by (Graves, 2013), In RNNs, an input sequence of variable length can be processed one element at a time while maintaining an internal memory state that stores information from the previous input. At each step, the current and previous inputs determine the output. As a result, RNNs have the advantage of storing sequence information and learning it, unlike other types of neural networks. A simple RNN architecture is shown in Figure 2.2, where ***A*** represents one chunk of the network, and a feedback loop allows information to be passed between the various stages of the network (Dautel et al., 2020).

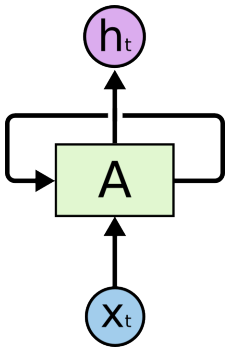

Figure 2.2: A Simple RNN Architecture.
Source: (Gupta et al., 2019)

Furthermore, Figure 2.3 depicts the RNN architecture in detail. It shows a network consisting of several duplicates, each passing a message to its neighbors (Gupta et al., 2019).

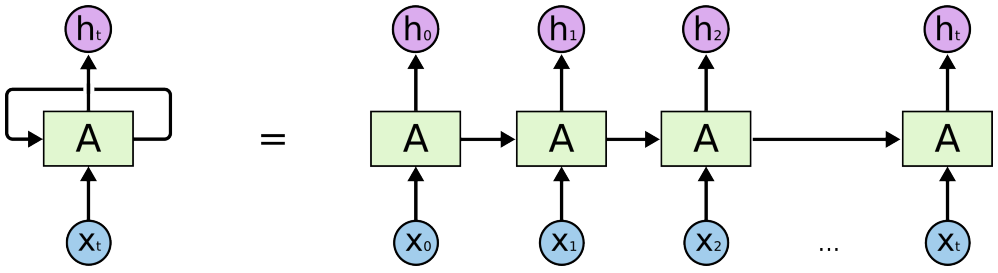

Figure 2.3: RNN Architecture in Detail.
Source: (Gupta et al., 2019)

RNNs can be used in various fields, especially in applications related

to NLP, such as machine translation and SA. However, RNNs suffer from the vanishing gradient problem, making it challenging for the network to recall long-term dependencies (Graves, 2013). In order to address this issue two types of RNNs are utilized, including Long Short-term Memory (LSTM), and Gated Recurrent Units (GRUs).

2.3.1 Long Short-term Memory

According to (Dautel et al., 2020), LSTMs are designed to store and output information using specific memory cells. These memory cells employ gates to control data flow into and out of the cells. As a result, they retain information for an extended time compared to a simple RNN, which explains LSTM's ability to address the vanishing gradient problem. Based on the input values and the previous state of the cell, the gates enable the network to store or forget information. Figure 2.4 shows LSTM architecture, depicting three gates in every network layer (Dautel et al., 2020).

1. The forget gate retains necessary information and discards unwanted information from the previous input.

2. The input gate learns new information.

3. The output gate passes the newly acquired information to the subsequent input.

In addition to gates, LSTM has two states: the long-term memory represented by $\boldsymbol{c}$ and the short-term memory represented by $\boldsymbol{h}$. LSTM processes the input through each layer and gate, learning to classify the data effectively (Dautel et al., 2020).

2.3.2 Gated Recurrent Units

GRUs are another type of RNNs that were created to handle the vanishing gradient problem. In comparison to LSTM, GRUs are a simplified variant (Rana, 2016). As explained by (Rana, 2016), GRUs combine the forget and input gates into a single gate called the update gate; this gate determines the important information to retain from past and current input. Furthermore, GRU has

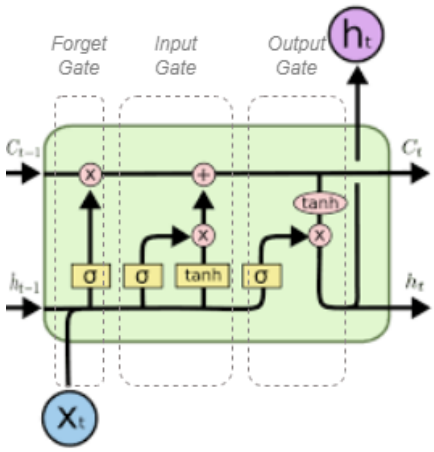

Figure 2.4: LSTM Architecture.
Source: (Dautel et al., 2020)

another gate called the reset gate, which determines which information should be discarded. This simple architecture makes GRU faster to run and easier to train (Rana, 2016).

2.3.3 Bidirectional Gated Recurrent Unit

As reported by (Abdelgwad et al., 2022b), a Bidirectional Gated Recurrent Unit (BiGRU) enhances the performance of a standard GRU by processing the input pattern in both forward and backward directions. The bidirectional nature of BiGRU enables it to capture past and future contextual information. It is beneficial in tasks where it is essential to understand a word or phrase's context to classify it accurately. (Fadel et al., 2022) states that by processing input in both directions, BiGRU can effectively consider the context surrounding each word or phrase, enabling DL models to make more accurate predictions.

2.4 Natural Language Processing

NLP is a branch of AI and Linguistics that teaches computers to understand statements or words written in human languages (Khurana et al., 2022). It uses computational techniques to automatically analyze and represent human language, motivated by theories of language and computation, as explained by (Chowdhary, 2020). However, the author also states that achieving a deep understanding of natural language by machines, on par with human understanding, is still a challenging goal yet to realize fully. NLP has multiple practical applica-

tions, such as online information retrieval, aggregation, and question-answering, which often rely on algorithms that operate on the textual representation of web pages; it plays a role in these applications to varying degrees (Chowdhary, 2020).

Moreover, most NLP methods rely on ML (Saireddygari, 2021). Including natural language generation and natural language understanding, which both advance the goal of text comprehension. NLP systems first break down unstructured data, such as social media posts, and pre-process it to create structured data that can be analyzed (Virmani et al., 2017). As further remarked in (Virmani et al., 2017), extracting information from the social network allows the structured use of a large amount of unstructured, distributed data.

2.4.1 Transfer Learning in NLP

It is the process of leveraging knowledge gained from one task to improve the performance of a model on a different but related task. The technique has recently gained popularity in NLP to overcome data availability limitations and enhance model performance. As part of transfer learning, pre-trained models trained on a large amount of data are used as a base for fine-tuning and adapting to new tasks with limited data. Additionally, widely used transformer architectures, such as BERT (Devlin et al., 2018) and GPT (Radford et al., 2019) , can also be utilized in transfer learning (Slovikovskaya, 2019). These architectures have been pre-trained on large text corpora and are ideal for transfer learning. As stated in the paper (Slovikovskaya, 2019) the integration of transfer learning and transformer architectures can significantly enhance the performance of NLP tasks.

2.4.2 Arabic NLP Challenges

Arabic NLP (ANLP) faces unique challenges due to the complexity and richness of the Arabic language (Habash, 2022). The morphology of Arabic is one of the most significant challenges for NLP systems. Words can take on various forms based on context, making it challenging to accurately identify and analyze each word's different forms. Furthermore, ANLP needs more standardized resources and annotated data than other languages, which hinders the development of accurate NLP systems. Additionally, Arabic has several dialects that differ significantly

in vocabulary and grammar, which poses another challenge for NLP systems in recognizing and interpreting text accurately (Habash, 2022).

These issues call for advanced solutions in order to develop Arabic-specific NLP systems (Habash, 2022). Researchers and practitioners in the field must develop robust methods to handle the morphology of Arabic, deal with the lack of standardized resources and annotated data, and account for variations in dialects. Such solutions would help develop accurate and efficient NLP systems for Arabic, thereby enhancing the quality of Arabic language technologies and supporting the development of applications that serve the needs of Arabic-speaking communities (Habash, 2022).

2.5 Text Mining

Text mining is a significant topic of study (Gupta et al., 2009). Text mining, described by (Inzalkar and Sharma, 2015), converts unstructured text into a structured format to find significant patterns and new perspectives. In addition, it aims to find previously undiscovered information that no one knows and has yet to be recorded. It overlaps with many fields, including data mining, NLP, statistics, and others. Text mining is a method that consists of various steps such as text preprocessing and text representation that allow for information extraction from unstructured text data (Talib et al., 2016).

2.5.1 Text Preprocessing

At this stage, textual input data is processed to remove all the noise apparent in unstructured text. Preprocessing involves transforming the text into a clean, consistent format that can be used to extract information from and classify it in a subsequent step (Muaad et al., 2022). NLP includes numerous techniques for text preprocessing. The following list describes some of the steps used in preprocessing:

- Text cleaning: includes the removal of punctuation marks, Arabic diacritics, repetitive unneeded letters in a word, and other symbols (Gamal et al., 2019).

- Tokenization: It breaks down text into a smaller structure called a token.

It can be words or characters (Kadhim, 2018).

- Stemming: A kind of normalization that results in a stem of the original word, for example, [علمتهن، متعلم ، تعلمنا] can be normalized to علم as explained by (Alasmari and Abdelhafez, 2022).

- Removing stop words: Words that are frequent in the text but do not hold a significant meaning. In Arabic, these can be [في، من، إلى]. Removing them reduces processing time and save storage in the dataset (Muaad et al., 2022).

2.5.2 Text Representation

Machines can instantly compute numerical data, but when it comes to textual data, they cannot process natural textual language directly. Therefore, after preprocessing, the text should be converted to numbers that can be computed (Muaad et al., 2022). This process is essential in text mining when following an ML approach and using NLP techniques. In ML, There are multiple text representation approaches; in this subsection, a few are listed:

- TF-IDF: Stands for Term Frequency-Inverse Document Frequency (Muaad et al., 2022). A model that can be implemented by the following phases:

  1. TF: Counting the occurrence of every word in the document or sentence, compared to the total of words in document or sentence.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}} \tag{2.1}$$

  2. IDF: Reflects the proportion of a word in a document or sentence compared to all words in entire corpus. It gives rare words higher

importance.

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents containing the term}}) \tag{2.2}$$

3. TF-IDF is the multiplication of these two phases scores.

$$TF - IDF = TF * IDF \tag{2.3}$$

As (Muaad et al., 2022) discussed, this process shows the words in the numerical form of a vector. The words that have more TF-IDF value or weight are considered rare and are emphasized by this approach. In contrast, common words are given lower weights.

- Word Embeddings: In (Bengio et al., 2003) the term "word embeddings" was coined. The proposed model in the paper was based on vectorizing words by training a neural language network. Embeddings are considered an effective tool, which encouraged the creation of several notable ones. The following lists a few embeddings models:
  - Word2Vec: Produces vectors in a vector space and maps them to words. Reveals word vectors that share a common or similar context and represent them close to each other in the vector space. Word2Vec utilizes both Continuous Bag of Word (CBOW) and skip-grams techniques (Mikolov et al., 2013).
  - FastText: It considers the internal structure of words or the sub-word information. (Altowayan and Elnagar, 2017) states that FastText gave promising results when applied to morphologically rich languages like Arabic.
  - BERT language model: Bidirectional Encoder Representations from Transformers (BERT) is an advanced language model that has significantly improved the performance of various NLP tasks (Devlin et al., 2018). In contrast to Word2Vec and FastText, which generate static word representations, BERT produces dynamic, context-aware word representations. To accomplish this, the model was trained on

two unsupervised tasks: masked language modeling and next sentence prediction. These tasks allowed the model to gain an understanding of sentence structures, grammar, and context (Devlin et al., 2018).

As a result of BERT's bidirectional architecture, it can take into account both left and right contexts simultaneously, leading to more accurate and richer word representations. With the model pre-trained on vast amounts of data, it can be adjusted for specific tasks with little additional training data, resulting in superior performance over traditional embeddings (Devlin et al., 2018).

2.6 Sentiment Analysis

SA, also referred to as opinion mining, is a field that utilizes text mining and NLP techniques to extract and analyze people's thoughts and feelings about specific things, which can be expressed in written text (Pozzi et al., 2016). This allows Analysts to learn more about individuals' attitudes toward various subjects, events, organizations, and products. With SA, public opinion can be tracked, consumer satisfaction can be measured, and it allows one to keep an eye on the company's reputation by understanding how people feel and behave toward services (Boudad et al., 2018).

There is often confusion about the difference between SA and opinion mining (Liu, 2020). SA and opinion mining concentrate on different facets of text analysis, but their underlying meanings are frequently connected (Liu, 2020). For example, أنا محبط للغاية من خدمة العملاء في هذا المنتجع, conveys a negative sentiment in an opinion about customer service. Opinions usually express positive or negative sentiments, but sometimes they do not (Liu, 2020). For example, هذا المنتزه هو وجهة شهيرة للسياح, is an opinion about the park, but it does not convey any sentiment. In such cases, the text would not contain any sentiment

information that could be extracted.

It is common for SA and opinion mining to be used to gain a complete understanding of the subjective. Concerning the fields' names, they are both extensively used in academia, but SA is primarily utilized in the industry (Liu, 2020). SA is particularly interested in discovering positive or negative sentiments, referred to as positive or negative opinions, in common speech.

2.6.1 Levels of Sentiment Analysis

The focus of SA research has been done at the three levels of classification (Boudad et al., 2018):

1. Document-level.
2. Sentence-level.
3. Aspect-level.

(Liu, 2020) explains that a document's overall sentiment is either positive, negative, or neutral. This classification level is the most popular and is suitable for figuring out the overall attitude of a document, such as a review or a post.

Whereas At the sentence-level, (Boudad et al., 2018) states that a document sentences are categorized according to emotion. It can help analyze the sentiment of specific document parts or the sentiment of several ideas contained within a single document. However, the sentiment of particular aspects of an item or service is classified at the aspect-level. For example, the sentiment about a restaurant's service is classified separately from the sentiment about its food (Liu, 2020). The aspect-level is explained in the following subsection.

2.6.2 Aspect-Based Sentiment Analysis

Contrary to the document and sentence levels in SA, Aspect-Based Sentiment Analysis (ABSA) introduces the aspect level. At this fine-grained level, parts of data are distinct which can be incredibly beneficial by giving more insight into the opinions. ABSA is classifying and analyzing the sentiment of text based on its aspects. For instance, الفندق اكثر من رائع من جميع النواحي in this statement, ABSA does not only classify the sentiment of the text as a whole. It first identifies

an aspect, then determines its sentiment. In our example, the aspect is الفندق, and the sentiment is positive.

In opinionated statements, aspects can be defined as an attribute or a part of an entity. Meanwhile, an entity is an event, person, service, or organization (Liu, 2020). For example, in the statement يوجد ضوضاء في الغرف المجاورة the entity is rooms الغرف, and the aspect is comfort. This decomposition of entities and aspects is helpful when an opinion is given on multiple aspects of the same entity. However, according to (Liu, 2020), in some fields, it might not be easy to distinguish between an entity and an aspect, or it is simply unnecessary to differentiate them. Furthermore, in the research field of ABSA, multiple tasks have been investigated. There are three main tasks that are frequently used. In the following, each is presented in detail.

- Aspect Term Extraction (ATE): In this task, aspects and entities are extracted from the text. (Liu, 2020) explained that entities are often omitted but are still crucial in analysis of aspects. Moreover, aspect terms can either be explicit or implicit. The former one stands for aspect terms that appear as noun or noun phrases. Such as, جودة الاكل ممتازة, the aspect food is explicitly defined. Whereas in the latter, the aspect terms are not noun or noun phrases. This can be seen in الحجز غالي, which implies the price aspect.

- Aspect Category Classification (ACC): Describes an aspect that falls into a category depending on the domain of interest. For instance, in hospitality domain, staff and employees aspects can be categorized into the service category (Liu, 2020).

- Aspect Sentiment Classification (ASC): Represents the classification assigned to either an aspect or an aspect category, which can be positive,

negative, and sometimes neutral (Liu, 2020).

It is worth noting that in literature, ABSA and its tasks can be expressed in several terminologies, including entity-based SA, target-based SA, and topic-based SA. The ABSA tasks terminologies used thus far are common in literature. Hence, they are adopted in the rest of this report to avoid confusion.

2.7 Sentiment Analysis Related Work

SA has been applied in many fields and domains, including business, hospitality, educational, and healthcare sectors. Thus, in healthcare and during the COVID-19 pandemic, some researchers and data scientists examined public opinion. The work cited in (Alhajji et al., 2020) concentrated on assessing the perceptions and attitudes of Saudi Arabian individuals regarding the country's preventive measures. The authors conducted this study by analyzing sentiments on Twitter, following a structured methodology. An annotated dataset was retrieved from Kaggle and preprocessed using stop-word removal, stemming, and tokenizing. Feature engineering was the next task, and it was accomplished using N-grams. Lastly, A NB supervised ML model was trained, establishing an accuracy and F-score of 89%. The model's success encouraged the researchers to employ it further and study people's sentiments toward every preventive event during the pandemic. This paper gave insight into how to follow an NLP methodology, aiding and providing an overall understanding of basic concepts. Whereas (Aljabri et al., 2021) analyzed public acceptance of the major preventive decision of shifting to distance learning. The authors limited this study to Saudi Arabia residents and only investigated Arabic tweets. Furthermore, data was collected and then preprocessed, followed by manually annotating the dataset. The paper explored and built multiple supervised ML classifiers and fed the input data represented in different features we also intend to utilize in this project, such as TF-IDF and N-grams. In addition, this study compared the ML models, revealing that Logistic Regression (LR) performed the best accuracy with 89%. (Alayba et al., 2017) have also conducted their study using multiple ML models and Deep Neural Networks (DNNs). Introducing their Twitter dataset that included healthcare services feedback. Tweets were collected for six months by researching well-

known hashtags in healthcare services; this shows similarity with our project data collection process described in chapter 3. In addition, the authors covered data filtration, preprocessing, and annotation techniques. Three individuals annotated the data as either positive or negative. Due to the difficulty of ranking opinions in Arabic, classifying tweets was only possible using two labels. Three ML algorithms were utilized: NB, LR, and SVMs with various settings. Word2vec was implemented with DL methods, which produced encouraging results of 90% for CNN and 85% for DNN. However, researchers claimed that SVM with Linear Support Vector Classification and Stochastic Gradient Descent delivered the best results. This paper encouraged us to try and use word embeddings in this project. Advances in Arabic SA have been incredibly noticeable. Scientists show interest and publish new approaches and frameworks, continuously investigating the field. (AlSalman, 2020) introduced a new approach to enhancing the classification of SA for Arabic tweets. The researcher used a publicly available dataset gathered from several topics, such as arts and politics. Data was operated on by processing the tweets using an N-gram tokenizer, a stemmer, and TF-IDF for feature extraction. This paper utilized the Discriminative Multinominal Naive Bayes model. The author evaluated the model using various metrics, which all revealed to outperform other mentioned related work in the paper. In Table 2.1 a summary is provided for these works

Table 2.1: SA Related Work

Research	**Aim**	**Dataset/Source**	**Arabic Language Type**	**Pre-processing**	**Approach**	**Features**	**Performance**
(Aljabri et al., 2021)	SA regrading shifting to distance learning.	Arabic tweets in healthcare /Twitter.	DA	Cleaning, Stop words removal, Normalization, Stemming, Tokenization	Supervised ML (SVM, Random Forest (RF), KNN, NB, LR)	N-Grams (Uni-gram, Bi-gram), TF-IDF	LR: Accuracy of 89.9% F-score of 89.9% Recall of 89.9% SVM: Precision of 94.5%
(Alhajji et al., 2020)	SA regarding COVID-19 preventive measures.	Arabic tweets sentiment in healthcare/ Kaggle.	DA	Cleaning, Stop words removal, Normalization, Stemming, Tokenization	Supervised ML (NB)	N-grams	Accuracy:89% Precision: 92% Recall: 86% F-score: 89%
(Alayba et al., 2017)	Public feedback about healthcare services.	Arabic tweets in healthcare /Twitter.	Not mentioned	Noisy data removal, Normalization	Supervised ML (NB, LR, SVM) DL (DNN, CNN)	N-Grams (Uni-gram, Bi-gram), TF-IDF	Accuracy: 90.14% for NB 88.32% for LR 91.37% for SVM 85% for DNN 90% for CNN
(AlSalman, 2020)	A new approach to enhancing the classification of SA for Arabic tweets.	Arabic tweets/Twitter.	Not mentioned	Tokenization, Stemming	Supervised ML (DMNB)	TF-IDF	Accuracy, Recall, F-score: 87.5% Precision: 87.6%

2.8 Aspect-Based Sentiment Analysis Related Work

Despite the challenges, some recent research has been published on ABSA for the Arabic language, indicating a growing interest in the field. This part takes a close look at the relevant documents to gain a better understanding:

- (Ashi et al., 2018) theorized in their work that pre-trained word embeddings can elevate the performance of the ABSA of Arabic tweets. This research utilized and compared word embeddings to perform two sub-tasks of ABSA, aspect detection and aspect sentiment classification. Following a text-mining methodology, the authors first collected Twitter data, particularly tweets related to the Saudi airline industry. Subsequently, data annotation was manually conducted on both aspect-level and sentiment level, revealing 13 aspects. Dataset was then fed to the two pre-trained word-embedding models (AraVec and fastText) to generate vectorized features. These features functioned as an input for the SVM models created for both ABSA subtasks mentioned previously. Results were promising, with accuracy rates of 70% for aspect detection and 89% for sentiment polarity detection, the fastText word embeddings model slightly outperformed the AraVec model.

- The researcher (Al-Ayyoub et al., 2017) suggested two DL models for the ABSA task's aspect-category identification and aspect-sentiment classification. They developed the first model for aspect-category identification by using CNN and stacked Independently Recurrent Long Short-term Memory (LSTM). The second model combines stacked bidirectional Independently Recurrent-LSTM with a position-weighting mechanism and numerous layers of attention mechanisms and is intended for aspect-sentiment classification. They used the Arabic SemEval-2016 dataset of hotel reviews to assess the proposed models. The baseline model and other models were outperformed by the first model, which had an F1 measure of 58.08%, and the second model, which had an accuracy measure of 87.31%

- (Al-Dabet et al., 2021) researched ABSA for Arabic. They focused on analyzing Arabic laptop reviews and explained how they created a dataset using the SemEval16-Task 5 annotation. The annotation process involved

predicting the aspect category and sentiment polarity label at both the sentence and text levels. This task aims to identify all targets in an opinionated review about an entity and determine the sentiment towards them. The researchers demonstrated how they built the dataset and used an SVM classifier. The results showed an accuracy of 73.2% in sentiment polarity, but low accuracy in predicting the aspect category, which had a precision of 52.9%, a recall with 22.5%, and an F1 measure of 31.5%.

- The researcher in(Alawami, 2016) investigated one sub-task of ABSA, which is aspect extraction. Explaining various approaches to conduct the task, the author aimed to analyze aspect extraction with a supervised method called Conditional Random Field (CRF). Due to that, the aspect extraction problem is treated as an information extraction task. Following a step-by-step procedure, restaurants reviews were collected using crawling tools. The collected dataset consisted of MSA and DA. Afterward, the labeling process was done with the assistance of two native Arabic speakers. The author then explained the CRF features chosen in this paper, including, tokenization, POS, and Stanford POS (SPOS) tagging. Lastly, to evaluate, the precision and the recall metrics were used, and the F- measure was employed as a comparison metric. Although they used various features, the results were essentially the same. However, using tokenazation and SPOS together has achieved the best performance, with a Precision of 68.30%, a Recall of 72.80%, and an F-measure of 70%. Table 2.2 summarizes this paper and the above mentioned papers in this part.

Table 2.2: ABSA Related Work

Research	**Aim**	**Dataset/Source**	**Arabic Language Type**	**Pre-processing**	**ABSA Task**	**Approach**	**Features**	**Performance**
(Ashi et al., 2018)	Compared two word embedding models for ABSA of Arabic tweets.	Airline service related tweets /Twitter	MSA, DA	Not mentioned	ACC, ASC	Supervised ML (SVM)	AraVec, fastText , N-Gram, Cosine similarity measure	ACC: Accuracy of 70% ASC: Accuracy of 89%
(Alawami, 2016)	Extract aspect terms from text written in Arabic dialects, with a focus on opinion mining.	Restaurant reviews / Jeeran	MSA, DA	Cleaning, Noisy data removal	ATE	Supervised ML (CRF)	Token, POS, SPOS	Precision of 68.30% Recall of 72.80% F-measure of 70%.
(Al-Ayyoub et al., 2017)	Build the Arabic Laptops Reviews dataset to focus on laptops reviews written in Arabic.	Arabic Laptops Reviews / Several websites	MSA, DA	Cleaning, Tokenization, Stemming, Stop Word Removal, Normalization	ACC, ASC	Supervised ML (SVM)	N-gram	ACC: Precision of 52.9% Recall of 22.5% F1-measure of 31.5% ASC: Accuracy of 73.2%
(Al-Dabet et al., 2021)	Improve the accuracy and effectiveness of ABSA for Arabic language texts by using deep learning techniques.	Hotels' reviews / SemEval-2016	MSA, DA	Not mentioned	ACC, ASC	DL (CNN, LSTM)	Not mentioned	ACC: F1-measure of 58.08% ASC: Accuracy of 87.31%

2.9 Related Tools

In the course of this project, we built the AI model using the Python programming language. Python is widely recognized as the most frequently used programming language in AI-related areas, including NLP. This section describes the tools and libraries that we utilized.

- Tools:
  - Visual Studio Code: It is a lightweight and cross-platform source code editor developed by Microsoft. It makes coding, debugging, and deploying software more productive by providing a wide range of built-in features. Visual Studio Code is also compatible with Git, making it a popular tool among developers (Microsoft, 2021).
  - Jupyter Notebook: Data scientists, researchers, and analysts use Jupyter Notebook to develop ML models and perform data cleansing, visualization, and analysis. It is easy to use, with a user-friendly interface and many pre-built libraries that make it suitable for both novices and experts (Jupyter, 2022).
- Python Libraries:
  - Natural Language Toolkit (NLTK): With NLTK, one can work with human language data and perform a wide range of NLP tasks, including tokenization, stemming, and part-of-speech tagging, all necessary for our project (Bird et al., 2009). However, NLTK's support for Arabic is less extensive than its support for other languages, such as English. NLTK features are not as advanced as those available in other libraries tailored to and designed for the Arabic language, like PyArabic and Farasa.
  - PyArabic: Is a library for Arabic text processing in Python. It includes modules for text normalization, word segmentation, and POS tagging, as well as tools for text classification, information extraction, and SA (Zerrouki, 2010).

- Farasa: Is an open-source library for ANLP. It offers various tools and features for pre-processing Arabic text, including tokenization, stemming, POS tagging, and named entity recognition. It is a well-established and actively maintained library written in the JAVA programming language but has been wrapped and used in Python (Abdelali et al., 2016).
- Scikit-learn: Several classifiers are provided by Scikit-learn that can be used for SA, such as NB, SVM, and Random Forest, among others (Pedregosa et al., 2011).

2.10 Conclusion

This chapter comprehensively reviewed the background information relevant to this project's problem. The review encompasses various domains, including but not limited to ML, DL, Transfer Learning, and NLP. Furthermore, the chapter delves into other research topics, such as the software tools, and related work in SA and ABSA. The chapter highlights the importance of thoroughly understanding the background information, data, tools, and techniques for addressing the problem. This understanding serves to clarify the project's idea and aids in the identification of the most appropriate approach for the project.

CHAPTER 3

DATA COLLECTION AND DATA REQUIREMENTS

Machine learning systems heavily depend on high-quality, abundant data for training, testing, and optimization. To build our web-based system, we require a domain-specific dataset, ensuring accurate and effective results. Twitter, widely recognized as a rich and real-time data source, was chosen for this project. This chapter outlines the data requirements (section 3.1), data collection process (section 3.2), data exploration (section 3.3), and data limitations (section 3.4), concluding in section 3.5.

3.1 Data Specification

Throughout this section, we define the textual data specification for our project and the two techniques used to guarantee the reliability and relevancy of data.

3.1.1 Twitter Data Specification

To complete the data acquisition process, the first step involved specifying the data's attributes to ensure the project's objectives are achieved. An examination of the field of study allowed us to identify common characteristics of Twitter data, including:

- The data should be textual, avoiding other types of data on Twitter, such as images and videos.
- The tweets that are gathered should only be in Arabic.
- Tweet textual content should be at least a sentence long to ensure that valuable information is extracted and analyzed.

- The tweets must be retrieved using keywords and hashtags associated with the entertainment industry to achieve relevancy.

3.2 Dataset Collection

An explanation of our dataset-gathering process is provided in this section. The primary purpose is to deliver a dataset that benefits the entertainment and events industry, directly supporting the tourism sector growth in Saudi Arabia. The dataset was gathered from public databases and by exploiting Twitter's Application Programming Interface (API).

3.2.1 Gathering Twitter Data

As defined by (Gupta and Gupta, 2019), an API is a gateway that allows access to a server's internal functions and allows users to interact and retrieve data by querying and accessing it programmatically. Numerous companies provide API services, such as Twitter. The platform provides developers with three access levels: essential, elevated, and academic research. First, we had to sign-up for a developer account, and then we could request elevated access to the API. A detailed form had to be filled out to explain how we intended to use the API. Afterward, we waited three weeks to get a response back approving our requested access. Four credential keys were provided for our project, allowing us to connect to the Twitter servers and retrieve all the data we needed to acquire. This data-gathering process depicted in Figure 3.1 had to be implemented in the Python programming language. Among the most popular and flexible approaches to data analytics is using Python libraries.

Several essential libraries were used in this step, including the Tweepy library for accessing the API and the Pandas library to save the fetched tweets into a Comma-Separated Value (CSV) file. These two libraries provide numerous features to operate with, but at this phase of our project, both were only utilized to collect data.

The process began by using the API search query to retrieve tweets from the main entertainment and events accounts, hashtags, keywords, and the comment section under certain events. During this process, a trial-and-error approach

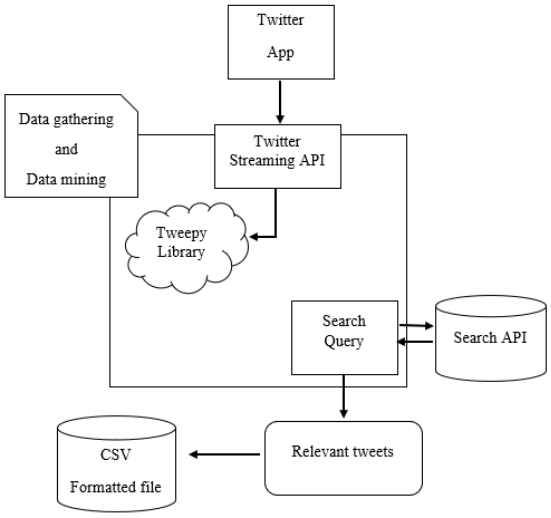

Figure 3.1: Data Gathering Proposed Approach.
Source: (Gamal et al., 2019)

was followed. Some of our data failed to fulfill this project's objectives, while other were successful. In the following subsections, we examine each search query in more detail.

3.2.1.1 Data Extracted by Accounts

The main accounts were selected manually; this was the beginning of our retrieval process. This approach focused on getting relevant tweets about the entertainment industry. As a result, Twitter handles that we extracted tweets from included, such as @GEA_SA, @Enjoy_Saudi, @RiyadhSeason, @Turki_alalshikh, @BlvdRuhCity, @JapanAnimeTown, and @ExperienceAlUla. Additionally, metadata such as location, count of retweets, and favorites was fetched. In our implementation, 300 tweets from each account were collected, for a total of 2100 tweets.

Inspecting the tweets introduced us to our first failed trial. A significant portion of the collected tweets were irrelevant to our project. Several tweets in these main accounts included advertisements which held no sentiment or opinion. Thus, irrelevant tweets were eliminated, and other methods of retrieval were explored.

3.2.1.2 Data Extracted by Events

Our first attempt at this method was to use the (Alshammari and AlMansour, 2020), extraction technique. The authors reported that they retrieved tweets about Saudi telecommunication companies but encountered the same issue of irrelevant data. As stated in the paper, the authors instead retrieved tweets from the companies' customer service Twitter accounts. We could not adopt their technique in our implementation since the entertainment and events sector only had the above-mentioned main accounts and no additional customer service accounts. Nonetheless, the paper prompted us to fetch data from mentions within tweets. Users can comment or reply to each tweet on Twitter by mentioning the tweet's author. Therefore, we decided to pick a few events and fetch data from the comment section. Each tweet in the comment or reply section has been retrieved individually and is separate in nature when saved into the CSV dataset file. In addition to ensuring relevance, this task provided us with many tweets that reflected opinions and held a sentiment. It also ensured that the tweets were up to date as they were retrieved during the event. Figure 3.2 displays the implemented Python code to collect the replies section of tweets.

```
# Authentication with Twitter
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api=tweepy.API(auth)
# fetching replies under a tweet using the account handle, and the tweet ID
accountName = 'Turki_alalshikh'
tweet_id = '1609670649982898176'
limit=300
columns =['User','Tweet','Location','FavoriteCount','RetweetCount']
replies=[]
for tweet in tweepy.Cursor(api.search_tweets,q='to:'+accountName, timeout=999999).items(1000):
    if hasattr(tweet, 'in_reply_to_status_id_str'):
        if (tweet.in_reply_to_status_id_str==tweet_id):
            replies.append([tweet.user.screen_name,tweet,tweet.user.location,tweet.favorite_count,tweet.retweet_count]

df = pd.DataFrame(replies, columns=columns)


df.to_csv('RepliesRiyadhSeason2.csv',encoding='utf-8-sig')
```

Figure 3.2: Code to Retrieve Replies of a Tweet.

Figure 3.3 Illustrates a Sample of Tweets Retrieved from the Replies Section of a Tweet Posted by @RyiadhSeason.

Figure 3.3: The Replies Section of a Tweet.
Source: (@RiyadhSeason, 2022)

3.2.1.3 Data Extracted by Hashtags and Keywords

Similarly, in this task, we retrieved tweets using hashtags, keywords, and phrases connected to our domain. Table 3.1 summarizes the keywords and hashtags we used to extract more data.

Table 3.1: Amount of Data Extracted from Each Hashtag and Keyword.

Hashtag	#بوليفارد_وورلد	300 tweets
	#انمي_تاون_اليابان	150 tweets
	#موسم_الرياض	290 tweets
Keyword	هيئة الترفيه	230 tweets
	مدينة العلا	30 tweets

3.2.2 Gathering Data from public datasets

In addition to collecting data using the API, we searched open-source databases such as Kaggle and found the Riyadh Season Twitter dataset. The data was gathered between the 16 and 21 of November in 2019. The dataset has several pitfalls, including data redundancy, such as spam tweets, and irrelevant data, for example, advertisements and event announcement tweets. Our team manually extracted tweets from the dataset to avoid the abovementioned problems.

3.3 Dataset Exploration

This section covers the dataset inspection and exploration. Throughout the data acquisition step, data exploration was conducted gradually on the dataset. In this part, our team identified some aspects of Saudi tourism that appeared in tweets, such as prices, services, attendees, weather, and safety. This can benefit us greatly in the data annotation step, as it showcases some of the labels in the dataset. In addition, by exploring the data, our team has recognized some limitations in the data collection process. These limitations include API restrictions and other limitations within the data in the dataset. Limitations are described in the following section.

3.4 Dataset Limitations

As with any data collection process, there are limitations. During this phase, our team encountered several issues that prompted us to change tools or eliminate certain data. In this part, we examine each of these factors:

- Twitter API Restrictions: The API only retrieves tweets posted in the past seven days, and it limits the number of requests we can make every fifteen minutes. To overcome this, our team utilized data scraping tools, such as Octoparse and Twitter Scraper by Apify. These tools provide various features to scrape data effectively.

- Data Irrelevancy: There are no restrictions on sharing data within hashtags or mentions on Twitter. As a result, the hashtags used in this project contained marketing tweets or tweets selling tickets to events. It became apparent that many tweets had nothing to do with the specific topic. The unrelated tweets shown in Figure 3.4 are excluded since we are mainly interested in opinionated statements.

- Multilingual Tweets: Despite filtering the language in our search query, the API and scraping tools still retrieved tweets written in other languages, mainly English. These tweets were excluded from the dataset.

1	العام_الجديد# طرب وموسيقى عروض متجولة فعاليات مختلفة أنشطة حماسية ومتنوعة تنتظركم في #بوليفارد_رياض_سيتي بأجواءها الحلوة احجزوا الآن https://t.co/vwq0h8u2Rw رأس_السنة_2023# بوليفارد_وورلد# https://t.co/PMAJ5gcNuO
2	اللي راحو بوليفارد سيتي وبنتظرون العد التنازلي لسنة 2023 كيف حالكم #بوليفارد_رياض_سيتي https://t.co/cfWS8PIY8t
3	تذاكر بوليفارد بوليفارد سيتي للبيع تواصلو خاص #بوليفارد_وورلد #بوليفارد_رياض_سيتي
4	زانت ليلة #تريو_نايت بمشاركة الفنّان الكبير جورج وسوف الغناء برفقة الفنّانين أنغام وبهاء سلطان في #بوليفارد_رياض_سيتي #موسم_الرياض https://t.co/b3G3iacFG7

Figure 3.4: Sample of Irrelevant Tweets.

3.5 Conclusion

In this chapter, we have clarified the data specifications adopted for the entertainment and events dataset and discussed the data acquisition process we followed to acquire the data. Additionally, we have highlighted some of the limitations of this dataset.

CHAPTER 4

SYSTEM REQUIREMENTS AND SYSTEM DESIGN

In this chapter, the system requirement analysis and design phase is introduced. This phase is essential in developing any software project, including our web-based tool for Aspect-Based Sentiment Analysis (ABSA). This chapter forms the foundation for the entire project implementation and is required to ensure its success. This phase aims to gather, analyze, and document the system's requirements and develop a complete design that meets these requirements. The data-gathering techniques used in this project are explained, followed by the requirements needed to build our system. Additionally, the initial design model of our system is discussed and illustrated in various diagrams. Lastly, the system prototype is revealed.

4.1 Information Gathering Techniques

In this project's system analysis and design phase, data gathering was performed using various techniques. These methods gave the development team critical insights into the problem area, ensuring stakeholders' needs were addressed effectively. These processes served as the foundation for establishing the system's requirements and identifying any limitations in the solution. For this project, interviews and questionnaires were chosen as the data collection methods. These have been successfully conducted and completed, significantly contributing to the project's subsequent phases.

4.1.1 Interview

In an interview, a researcher or an interviewer prepares a list of questions to ask stakeholders during the system's analysis phase. In this conversation, the questions are usually open-ended to allow for more in-depth data to be gathered. Interviews are an effective method to obtain knowledge of the problem area from a stakeholder's perspective. In this project, our team was motivated to interview at least one small or start-up company in the entertainment tourism industry. Our team contacted a travel company called Pangaea and requested an interview. Pangaea is a company and a club with a vision of pioneering adventurous and Eco-tourism activities all over the Kingdom. The interview was with the company's marketing consultant; the interview questions are seen in Table 4.1.

Table 4.1: Interview Questions Summary Table.

Interview Questions
Q1: What is your job position?
Q2: Explain Pangaea's vision in relation to the tourism industry.
Q3: Can you describe the company's current process for monitoring and analyzing customer feedback?
Q4: Does your company keep track of clients' feedback on social platforms, such as Twitter or Instagram?
Q5: What challenges are faced when trying to gather feedback or conduct the analysis?
Q6: How important is it for your company to have an accurate understanding of your client's sentiments about the activities and services provided?
Q7: What precisely do you need in an automated system?

During this interview, the consultant provided valuable insights into how most small businesses in the industry operate regarding client feedback. The interviewee stated he worked as a marketing consultant for various companies, including Pangaea. When asked about Pangaea's current feedback monitoring process, he said that clients who book activities with them, for instance, exploring Alula, are given surveys afterward to express their level of satisfaction. These surveys are then manually processed by the customer service team and analyzed and visualized manually as well. Additionally, the consultant was explicitly asked about tracking feedback on social platforms. According to him, a small team tracks posts or tweets and tries to assess the overall sentiment. The

consultant explained that the company is aware of this approach's challenges, saying it consumes time and resources. He stated that although data analytics businesses provide helpful automated tools, the paywall is a significant issue for small companies in the field. Lastly, the consultant was asked about any suggested design features for our automated tool. As expressed by him, such tools should have an easy-to-use interface, especially for novice users, since it will help start-up companies save time and training resources.

4.1.2 Questionnaire

This questionnaire aimed to gain insight into the opinions and preferences of those in the tourism industry regarding using Twitter for gathering information and feedback. In this questionnaire, we aimed to explore the problem area from two main perspectives and assess the need for our tool. The questionnaire was conducted using Google Forms and received responses from 90 participants, including 72 tourists and 18 organizations. Both types of participants had access to the same questionnaire with slightly different questions for each, but still similar. The results highlight the need for a web-based tool for analyzing public opinions on specific aspects of Saudi tourism. The questionnaire consisted of seven different questions asked to individuals and organizations. Respondents were asked structured questions such as ranking scale, checkbox selection, and yes or no questions. The list of questions is summarized in tables 4.2. The analysis of the answers are included in Appendix I.

Table 4.2: Questionnaire Summary Table.

Question
1. Do you utilize Twitter for researching and gathering information about tourism and entertainment destinations or events?
2. On a scale of 1-3, how important is it for you to know the public opinion on destinations or events before visiting them?
3. On a scale of 1-3, how difficult do you find it to obtain accurate feedback and opinions of others about Saudi tourism destinations and events?
4. On scale of 1-3, how satisfied are you with traditional methods (e.g., reading tweets) to find what people think of the tourism industry?
5. Would you like to use a web-based tool for analyzing public opinions based on specific aspects of Saudi tourism on Twitter?
6. What aspects of tourist or entertainment destination or event are most important to you when considering visiting?
7. What features would you like to see in this tool?

4.2 Requirements Specification

The requirement specification stage served as a guide for the development of our system. It clearly defined the system's functionality and expected behavior, facilitating a mutual understanding of expectations between stakeholders and the development team. By creating a detailed and comprehensive requirement document, we ensured the developed system meets the end-users' needs. The requirements collected included functional, non-functional, and data-related aspects, all of which are explained and outlined in this section.

4.2.1 Functional Requirements

The purpose of this part is to introduce a set of functional requirements that specify the various tasks that the system must be able to handle. The requirements emphasize what tasks the system should perform rather than how it implements them (Kung, 2013). The following list concludes our system's functional requirements:

FR.1: The system shall allow the user to create an account.

FR.2: The system shall allow the user to log in and log out.

FR.3: The system shall allow the user to upload a file of tweets.

FR.4: The system shall allow the user to view their history records.

FR.5: The system shall be able to perform ABSA on the tweets using the Deep Learning (DL) models.

FR.6: The system shall be able to display tweets with ABSA results on the dashboard.

FR.6.1: The system shall be able to display the text of the tweets.

FR.6.2: The system shall be able to display the aspects and the sentiment of each aspect next to the displayed tweet.

FR.7: The system shall allow the user to visualize the results using plots or other visual representations, such as word clouds, pie charts, and bar charts.

FR.8: The system shall allow the user to filter tweets by sentiment, either positive or negative.

4.2.2 Non-Functional Requirements

Non-functional requirements define the system's quality attributes, such as its usability, reliability, scalability, and performance (Kung, 2013). In this part, we list our system's non-functional requirements.

NFR.1: The system should be accessible to all users 24/7.

NFR.2: The system should support the Arabic language.

NFR.3: The system should be easy to use.

4.2.3 Database Requirements

The database in our system is used to store the history records of the users' analyses. During the process of determining system requirements, data requirements is specified in order to design and construct the system's database. The tables 4.3, 4.4, 4.5, 4.6, and 4.7 summarize the data and include an example column to illustrate our system's database requirements.

Table 4.3: Database Requirements: User.

User		
Data Field	Description	Example
UserID	A unique key for every user.	9
Fname	The user's first name.	Asma
Lname	The user's last name.	Ali
Email	The user's registered email.	AsmaAli@gmail.com
Password	The user's chosen password, which must be encrypted to establish security.	656000$EuxtFJnx

Table 4.4: Database Requirements: Aspect Sentiment Data.

Aspect Sentiment Analysis		
Data Field	Description	Example
ASID	A unique key for every aspect and sentiment Data.	7
Count	Count of the combined pair of aspect and sentiment.	20

Table 4.5: Database Requirements: History.

History		
Data Field	Description	Example
HistoryID	A unique key for every history record.	5
Date	Stores the date of analysis.	09-Jan-2023
NumOfTweets	Stores the number of tweets uploaded by the user.	16

Table 4.6: Database Requirements: Aspect.

Aspect		
Data Field	Description	Example
AspID	A unique key for every aspect.	7
Name	Name of aspect found in tweet text.	الخدمة

Table 4.7: Database Requirements: Sentiment.

Sentiment		
Data Field	Description	Example
SentID	A unique key for every sentiment.	7
Type	Type of sentiment.	Positive

4.2.4 Software Requirements

The developed system is a web-based tool compatible with a variety of web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. The tool was built using multiple software tools and libraries as detailed in section 2.9.

4.2.5 Hardware Requirements

Our solution was designed with user accessibility in mind. It requires the user's hardware to have an internet connection, the ability to run a web browser, and a stable internet connection for optimal usage.

4.3 Initial Design

4.3.1 Entity-Relationship Diagram

An Entity-Relationship (ER) diagram provides a visual representation of a database design. It illustrates the relationship between entities and attributes in the system's database and is fundamental to the development of an efficient database (Elmasri, 2021). ER helps recognize potential system design flaws, such as data redundancy, and avoid them. As shown in Figure 4.1, the ER diagram of our system has five entities: User, HistoryRecord, AspectSentimentAnalysis, Aspect, and Sentiment.

In the following step, the ER diagram is modeled into a relational database schema using data mapping. Relational schema provides a template for organizing data in a relational database, ensuring that data is stored consistently and logically, making querying the data easier. Additionally, normalizing the schema is necessary to eliminate null and redundant values. Figure 4.2 shows our system's relational schema after normalization.

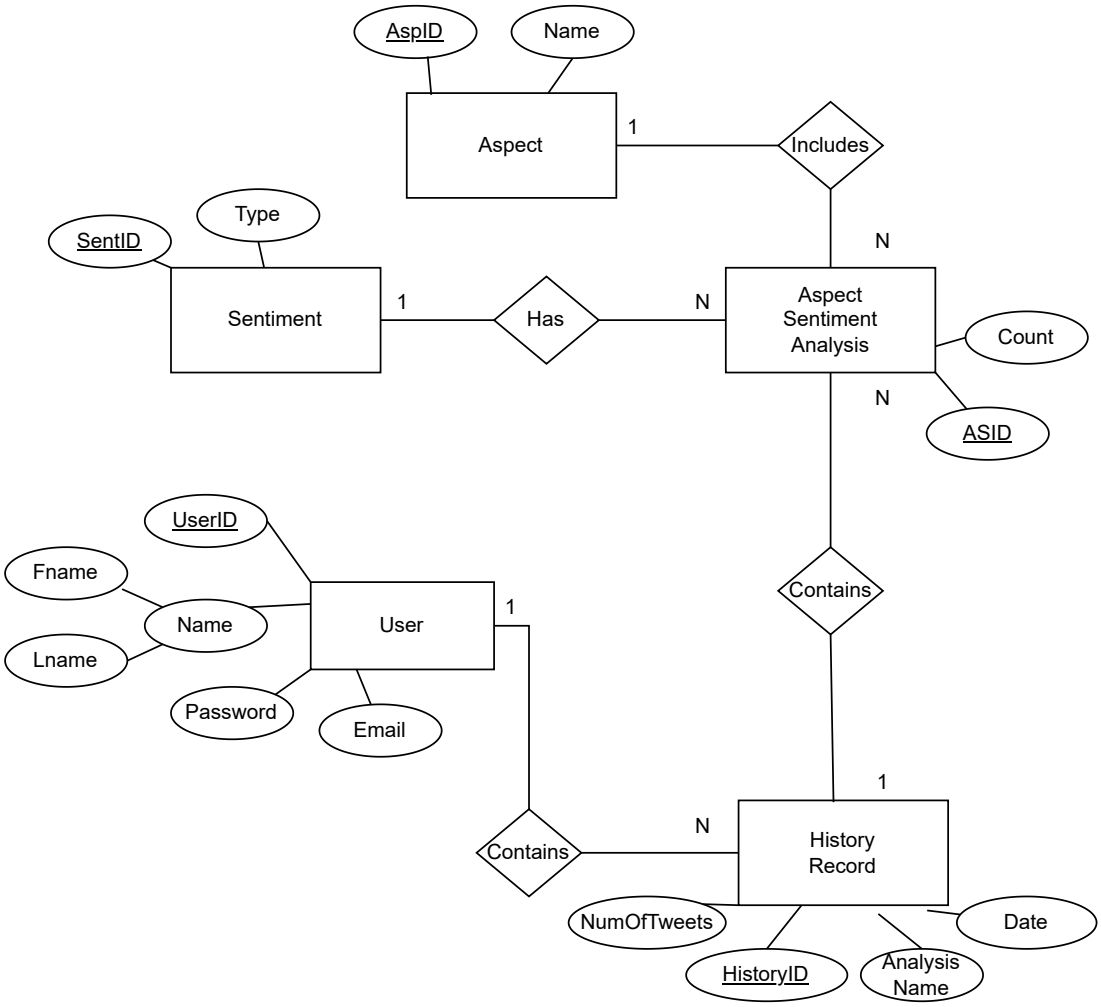

Figure 4.1: The Entity-Relationship Diagram of the System.

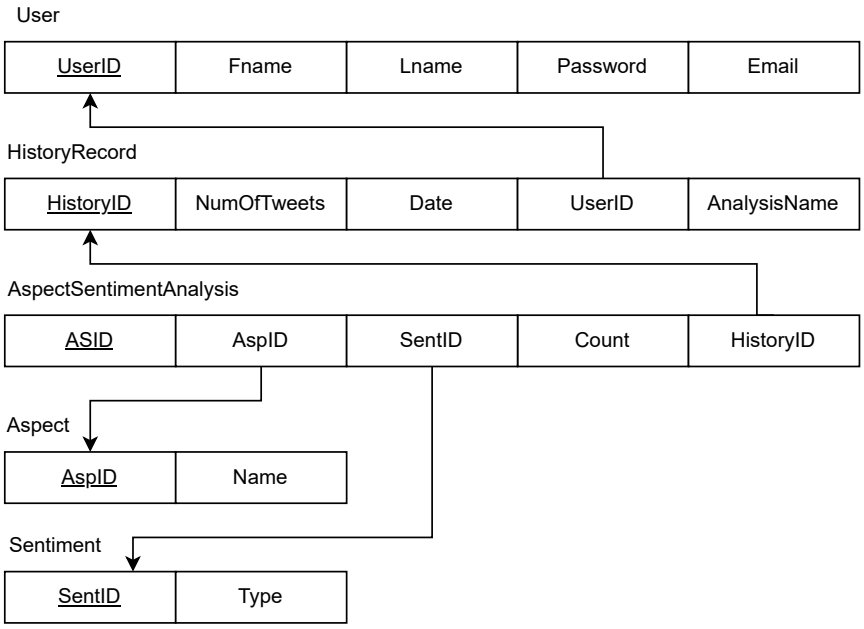

Figure 4.2: The Relational Database Schema of the System.

4.3.2 Use Case Diagram

A use case diagram is produced during the system analysis phase to represent the interactions between users and a system, showing the different actions end-users can do with the system (Kung, 2013). Figure 4.3 depicts our system's functional requirements in use cases, with the actor being the user. The user can perform a variety of tasks in the system. In addition to the use case diagram, this part also describes the most important use cases in tables 4.8 to 4.11.

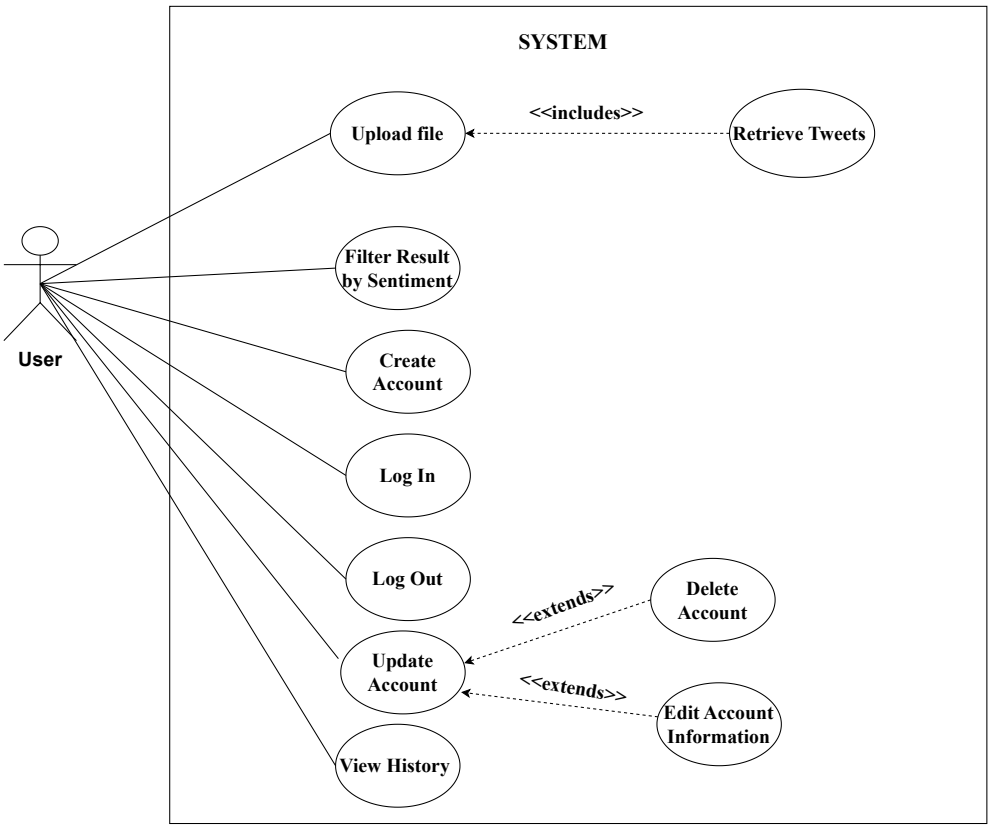

Figure 4.3: The Use Case Diagram of the System.

Table 4.8: Use Case Description: View History.

Use Case	View History
Brief Description	users can view their history details.
Actors	User.
Pre-condition	The user logged into their account.
Basic Flow of Events	1- The user clicks the view history option. 2- The system displays the user analysis history.
Post-condition	The history displayed.

Table 4.9: Use Case Description: Upload File.

Use Case	Upload File.
Brief Description	The user can upload a file and submit it for analysis.
Actors	User.
Pre-condition	The user logged in.
Basic Flow of Events	1- The user selects a file to upload. 2- The user uploads the file for analysis. 3- The system stores the file in a temporary directory for analysis.
Post-condition	The system sends the file path for text retrieval and stores it in the temporary directory simultaneously.

Table 4.10: Use Case Description: Retrieve Tweets.

Use Case	Retrieve Tweets
Brief Description	Followed by uploading a file by the user, the system begins retrieving tweets, then performs ABSA.
Actors	User.
Pre-condition	The user uplaoded the file for analysis.
Basic Flow of Events	1- The system starts retrieving the text data from the file then sends it to the ABSA model. 2- The ABSA model receives the tweets data. 3- The model analyzes the tweets, extracting aspects and sentiments. 4- The system displays the tweets with extracted aspects and sentiments on the dashboard.
Post-condition	The result is visualized to the user.

Table 4.11: Use Case Description: Filter Results by Sentiment

Use Case	Filter Results By Sentiment
Brief Description	The user can filter the the results by a chosen sentiment.
Actors	User.
Pre-condition	The user has uploaded a file and received results.
Basic Flow of Events	1- The user clicks on a filtered tab (positive or negative or neutral). 2- The system displays the tweets, aspects, and sentiment result filtered.
Post-condition	The filtered results by sentiment are displayed.

4.3.3 Class Diagram

Class diagrams are commonly used in object-oriented design and development to depict the system's structure and to verify the design against requirements. Class diagrams can also aid in the implementation phase by using them to generate code. In Figure 4.4, our system's class diagram is presented.

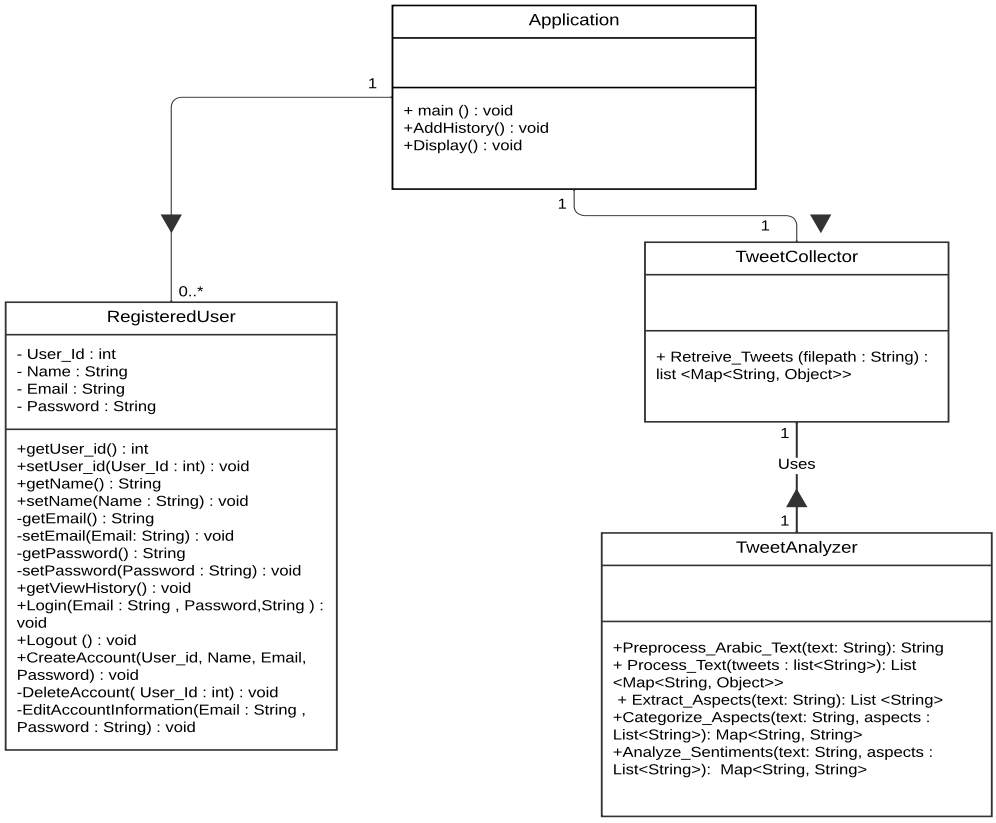

Figure 4.4: The Class Diagram of the System.

4.3.4 Sequence Diagram

Sequence diagrams illustrate the interactions between components in the system over time and can also be utilized to model the flow of control in the system (Kung, 2013). A sequence diagram is an excellent representation to communicate the design of our system. Figure 4.5 to 4.10 shows each sequence diagram for some use cases of our system.

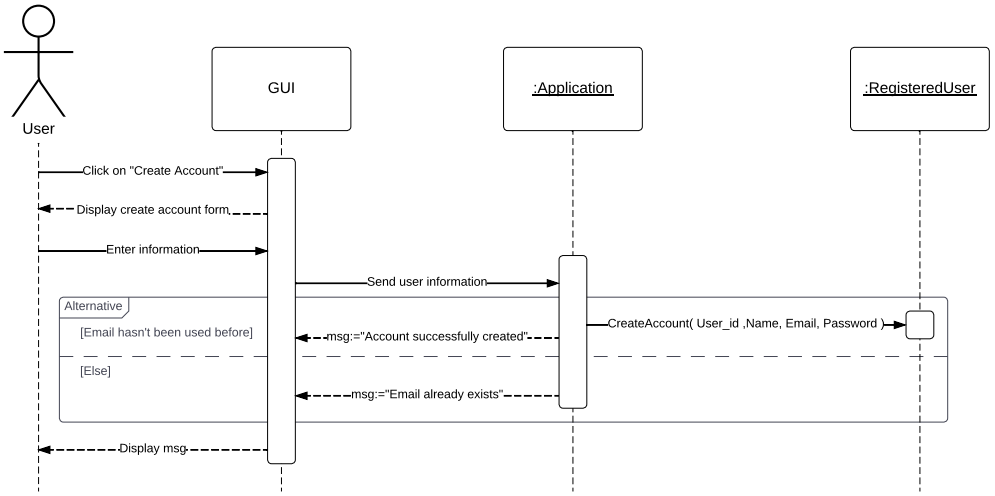

Figure 4.5: Sequence Diagram for Create Account.

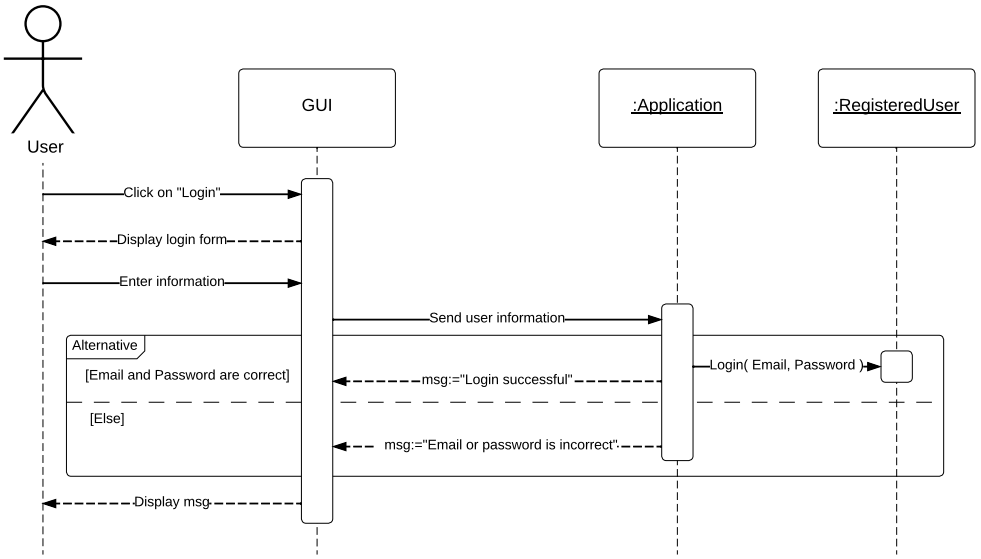

Figure 4.6: Sequence Diagram for Login.

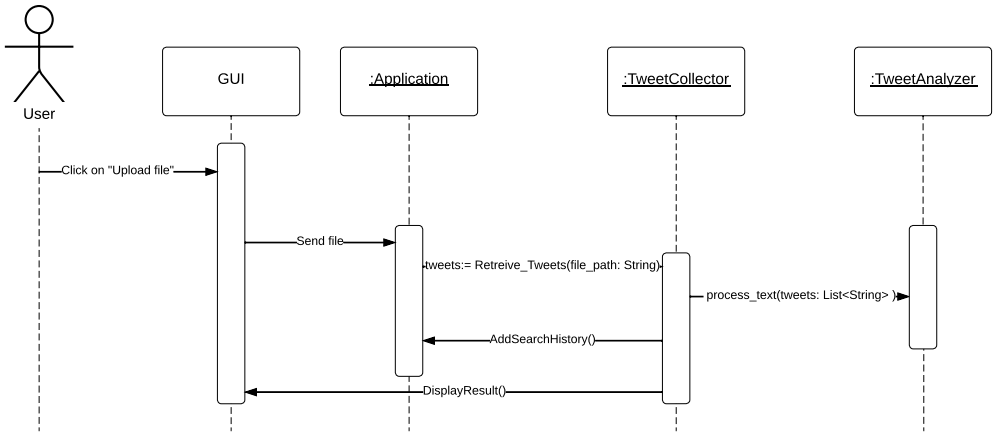

Figure 4.7: Sequence Diagram for Upload File.

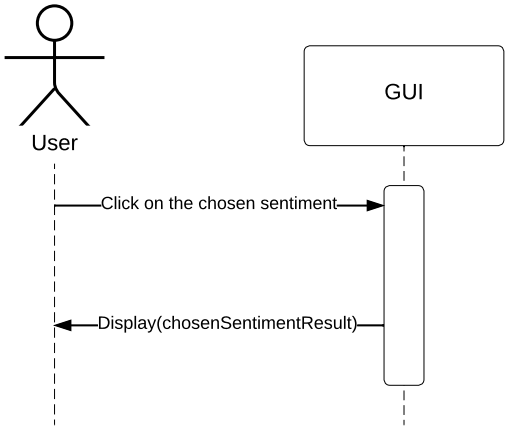

Figure 4.8: Sequence Diagram for Filter Tweets by Sentiment.

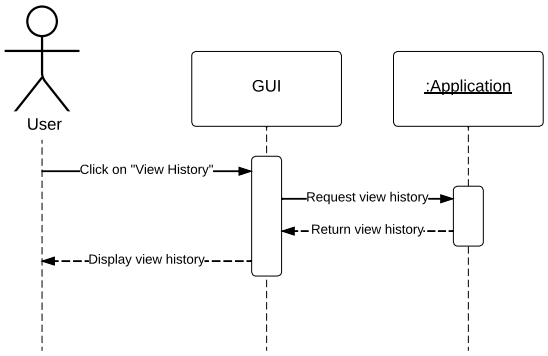

Figure 4.9: Sequence Diagram for View History.

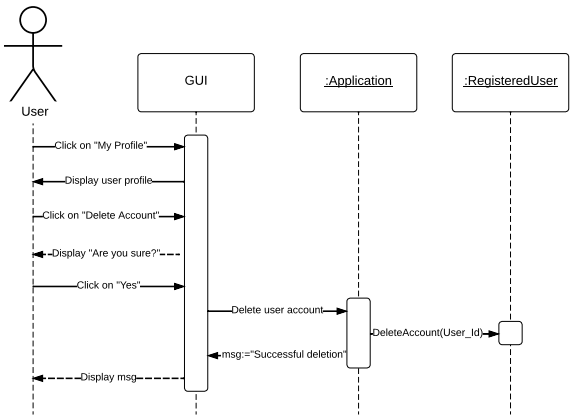

Figure 4.10: Sequence Diagram for Delete Account.

4.3.5 Design Modeling Tools

In any system's design phase, developers usually utilize modeling tools that provide various features to easily and quickly draw the system's diagrams. In this section, a few web-based modeling tools were used, such as Draw.io for illustrating our use case diagram and Lucidchart for the rest of the diagrams.

4.4 Prototype

An initial prototype of a web-based tool involves constructing a sample of its design. Before proceeding with final development, this step is essential in the development process as it enables the development team to evaluate designs, implement changes, and enhance the overall concept (Kung, 2013). Prototypes also give stakeholders a better view of how the tool operates and look. In this section, our system's prototype is presented.

4.4.1 Interface Type

The prototype is medium-fidelity designed to be implemented via a web interface for optimal cross-platform access using different web browsers. The prototype has a user-centered design intended to meet the expectations of end users who commonly utilize web-based applications for data analysis. Additionally, it displays all functionalities related to the user and presents a clear demonstration of the information architecture within the interface.

4.4.2 Interface Description

The main prototype interfaces are shown in this part, and each function is described. First, our system's main interface is displayed in Figure 4.11.

Figure 4.11: Main Interface Layout.

Figure 4.12 displays the sign-up layout for end users. After creating an account, if the user session terminates, they can log in to the system again, as seen in Figure 4.13.

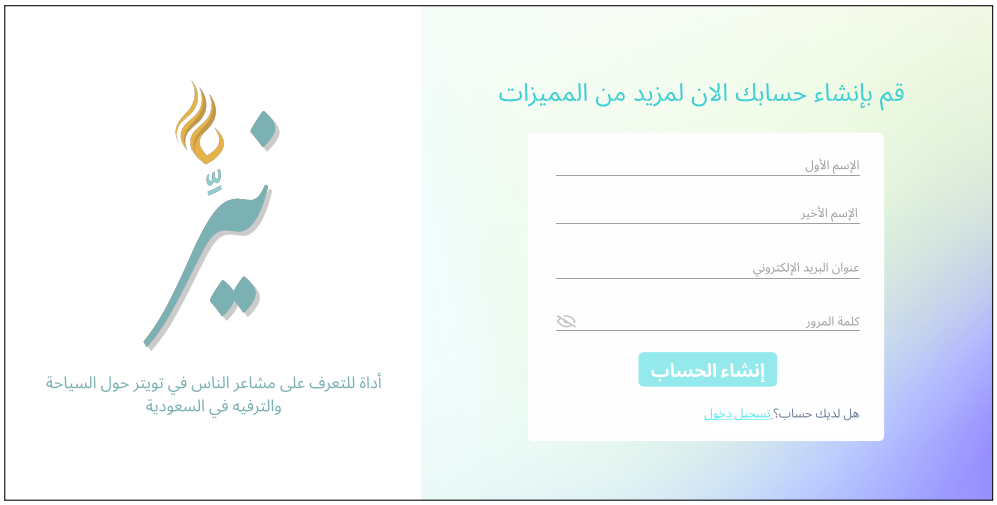

Figure 4.12: Sign-up Interface Layout.

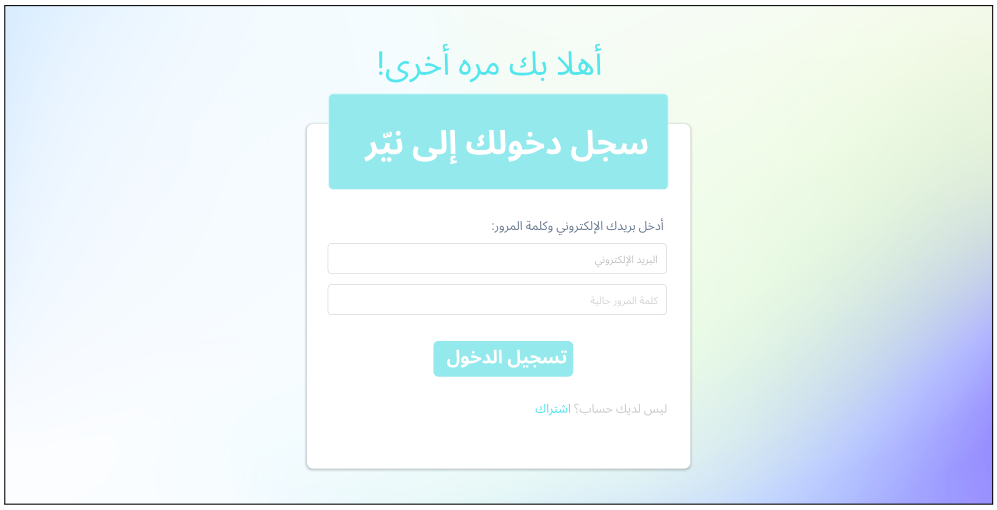

Figure 4.13: Log-in Interface Layout.

Users are presented with the upload file interface in Figure 4.14. In the upload file interface, users can choose a file to perform ABSA on it and receive results visualized in four charts. As seen in Figure 4.15, the tweets menu displays tweet text and the aspect and sentiment found next to the tweet. Additionally, the main dashboard visuals include aspects of tweets, sentiment of tweets, sentiment by aspect, and a word cloud.

Figure 4.14: Upload a File Interface Layout.

Figure 4.15: User Dashboard Interface Layout.

Users can also view their history, as displayed in Figure 4.16, and can display any record content on the dashboard as seen in Figure 4.17.

Figure 4.16: User View History Interface Layout.

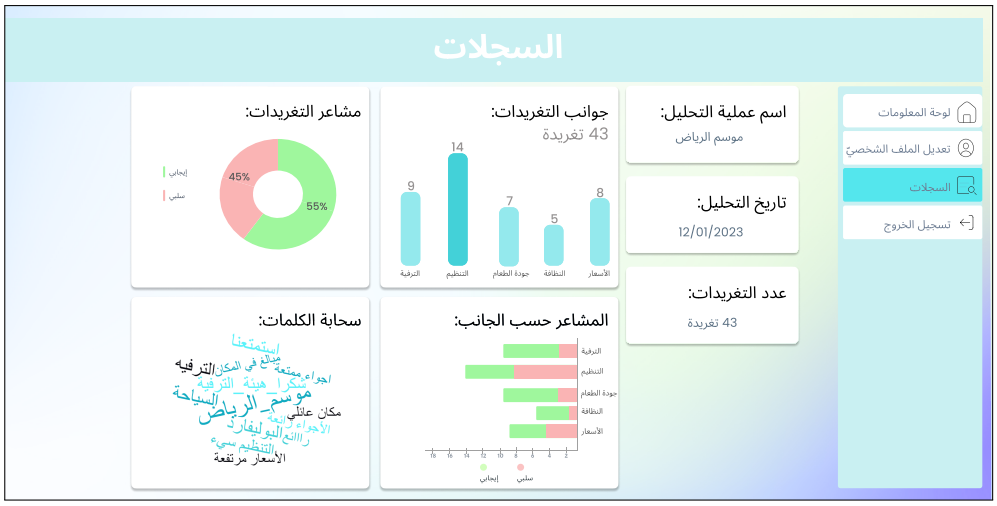

Figure 4.17: User View History Dashboard Interface Layout.

In addition, the user's personal information can be accessed, edited, or deleted, as seen in Figure 4.18. A confirmation message is generated if the user selects to delete their account, as shown in Figure 4.19.

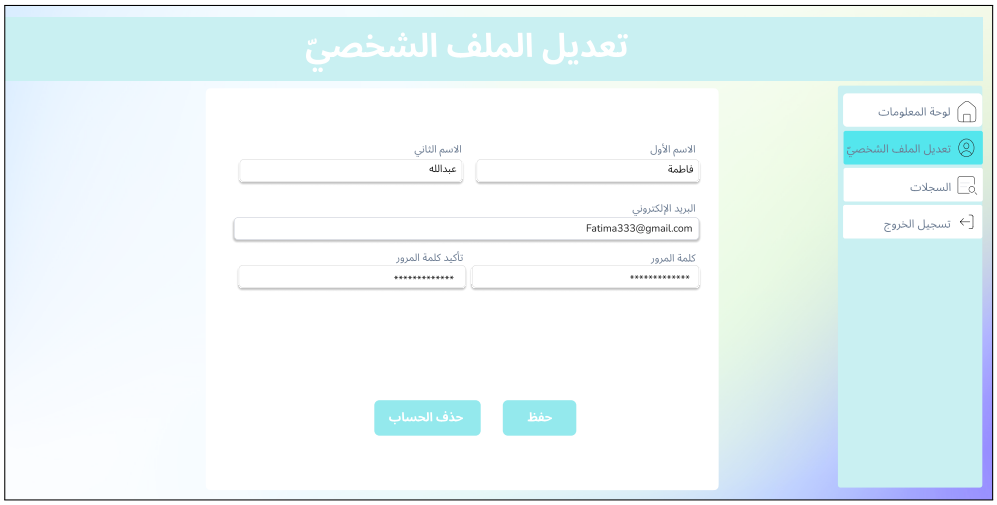

Figure 4.18: User Profile Interface Layout.

Figure 4.19: User Confirmation Message for Account Deletion.

4.4.3 Prototype Design Tools

The prototype in this section was constructed first using a sketch then our team used Figma (Figma, 2022) as the selected prototyping tool. Also, Adobe Photoshop was used to design the logo for the system.

4.5 Conclusion

The development of the suggested system software was discussed in this chapter. It highlighted the crucial steps in the development, including determining the requirements for the system. The design of the system was also discussed, with the use of diagrams such as a use case diagram, class diagram, sequence diagram, entity-relationship model, and database schema to illustrate key aspects of the design. Additionally, the system's initial prototype was constructed and revealed.

CHAPTER 5

DATASET

This chapter begins with a brief description of the previous work, which covers different completed phases in this project. The rest of the chapter explains the annotation of our entertainment and events dataset collected by this project's team, as discussed in Chapter 3. This chapter also reveals the obstacles encountered during the annotation process and the choice of an alternative dataset to overcome these challenges.

5.1 Previous Work Overview

In order to develop Nire, we conducted a comprehensive review of existing literature and studies on Aspect-Based Sentiment Analysis (ABSA) algorithms and their applications. This involved examining various ABSA tasks, including aspect extraction and aspect sentiment classification, to identify our project's most suitable algorithms and approaches.

Since our project focused on analyzing the sentiment of tweets related to tourism in Saudi Arabia, obtaining a dataset was a critical component of our ABSA system. However, a suitable dataset was not available, so we had to develop a method to collect it. our team scraped tweets from Twitter using specific keywords related to Saudi tourism. Then we thoroughly reviewed the collected data, followed by extensive pre-processing to eliminate irrelevant tweets and reduce noise.

In addition to our literature review and dataset collection efforts, we also conducted surveys and interviews with individuals and organizations familiar with the tourism industry in Saudi Arabia. This allowed us to better understand their

needs and analyze requirements and helped us implement an initial design of the project requirements and prototype.

5.2 Dataset Annotation

This section describes the dataset annotation process for aspect term and aspect sentiment, applied to the Arabic entertainment and events dataset. The dataset consists of 3,250 tweets, while the annotation process involved two steps: aspect term annotation and aspect sentiment annotation.

Four native Arabic speakers, two undergraduate students who volunteered their expertise in the study field, and two project team members were chosen as annotators to ensure diverse perspectives. The annotators received comprehensive training and clear annotation guidelines, found in Appendix II, ensuring consistent and accurate results.

The annotation process excluded tweets without aspect terms or sentiments, resulting in a final annotated dataset containing only relevant tweets with clear aspect terms and sentiments. The entire dataset annotation process was completed within a three-week timeframe.

Quality control measures were employed to ensure accurate and consistent annotations. These measures included regular meetings, spot checks, and feedback sessions. The annotation process resulted in a final dataset of 1,200 tweets with aspect terms and sentiment polarities labels.

5.3 Annotation Challenges

In this project, the manual annotation of Twitter data posed several challenges. Our team initially spent months gathering relevant tweets, cleaning the data, and preparing it for annotation. Despite these efforts, we faced several obstacles that impacted our progress.

- **Aspect category annotation:** We realized the importance of aspect category in our dataset after the annotation process, as existing literature on ABSA often focused on only two tasks per study, neglecting a comprehensive overview of all tasks involved. Thus, we added an aspect category annotation step but faced time constraints that prevented us from

completing this task.

- **Insufficient tweet data:** The limited dataset of 1,200 tweets posed a challenge for training the ABSA models due to the complexity of Arabic language and sentiment classification difficulties. Larger annotated datasets are needed for robust models.

- **Time management:** Meeting deadlines was challenging due to the time-consuming annotation process.

These challenges significantly impacted our project's progress, as they affected the quality and quantity of the annotated dataset, which adversely affect the performance of the ABSA models. As a result, it became clear that our project required an alternative solution to address these challenges and meet our objectives.

5.4 Alternative Dataset

Recognizing the importance of time management and adaptability, our team chose the Arabic hotel reviews dataset as an alternative to our original tweets dataset (Mohammad et al., 2016). Created for SemEval-2016 Task-5, this dataset is designed for Arabic ABSA tasks and provides a rich source of annotated data.

The dataset comprises 6029 reviews, divided into 4802 for training and 1227 for testing. These Arabic hotel reviews were initially annotated with aspect terms, aspect categories, and aspect sentiment labels. Table 5.1 presents a sample of these original annotations from the dataset. Subsequently, our team reused and relabeled the aspect terms to better serve the requirements of our models. The detailed process of this reannotation is discussed in Chapter 6. As for the aspect categories, our team relabeled them to align with other ABSA datasets, as explained in the upcoming subsections.

Table 5.1: Samples of Sentences from the Arabic Hotel Reviews Dataset.

Review Sentence	Aspect Terms	Aspect Categories	Aspect Sentiments
أوافق.. يحتاج تجديد شامل كان افضل منتجع في مدينة الخبر، الموقع رائع ولكن المرافق سيئة للاسف	الموقع المرافق	LOCATION#GENERAL FACILITIES#GENERAL	Positive Negative
فندق رائع ولكن الخدمة ضعيفة جميل وخلاب ويقع في وسط المدينة	فندق الخدمة	HOTEL#GENERAL SERVICE#GENERAL	Positive Positive
تصميم الغرف رائع ومساحتها الواسعة وحداثتها من حيث الاثاث	الغرف الاثاث	ROOMS#GENERAL ROOMS#QUALITY	Positive Positive

5.4.1 Aspect Terms

In the review text, aspect terms refer to specific words or phrases that relate to various aspects of the hospitality domain, such as the staff, the service, the food, or the location. By identifying these terms, we can determine which aspects of the hotel experience are being discussed.

5.4.2 Aspect Categories

Aspect categories in the dataset helped organize the aspect terms into broader, more manageable groups. By examining the dataset and its distribution of aspect categories, we could better understand which aspects of hospitality domain were most commonly discussed in the reviews.

The dataset initially contained 34 categories, labeled using the {Entity# Attribute} annotation scheme adopted from the SemEval-2016 annotation guidelines (Pontiki et al., 2016). In this scheme each aspect term was associated with an entity and an aspect attribute. The entity represented the subject, while the aspect represented a specific characteristic or attribute related to the entity. For example, for the sentence "The food was cold," the entity would be "Food," and the aspect would be "Quality," which resulted in the Food#Quality category as displayed in Figure 5.1.

Figure 5.1: Example Sentence from the Arabic Hotel Reviews Dataset.

However, our team observed that many categories appeared infrequently as seen in Figure 5.2.

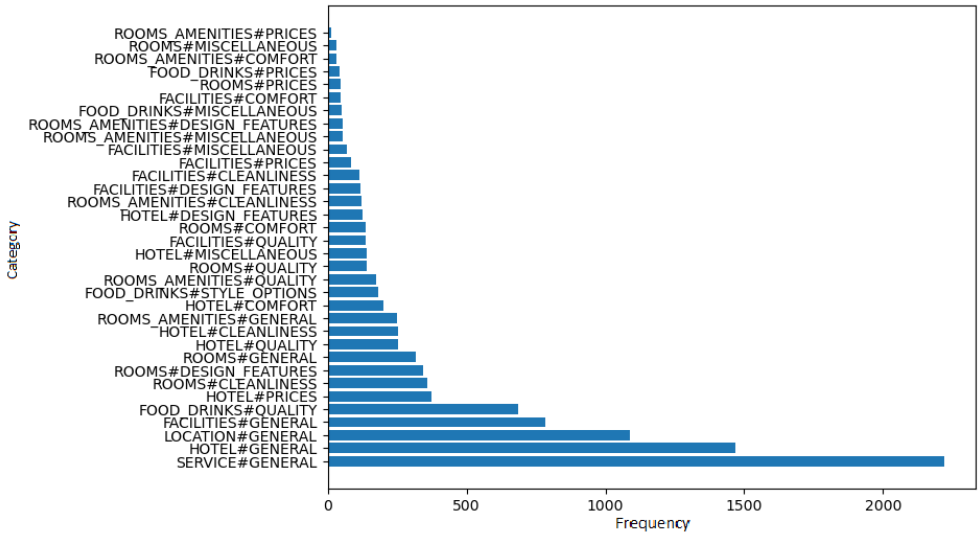

Figure 5.2: Categories Distribution in the Arabic Hotel Reviews Dataset.

Insufficient data for these categories can cause problems during model training and result in poor performance. Therefore, our team decided to relabel the categories to consider only the entity part, effectively grouping aspect terms into broader categories such as service, food, and location. This approach was adopted following the annotation scheme of other Arabic ABSA datasets, such as human annotated dataset of book reviews (Al-Smadi et al., 2015) and the Arabic news dataset (Mohammad et al., 2015). These datasets only considered the categories as one word entity.

Furthermore, Table 5.2 provides a summary of the newly adopted aspect categories, along with a brief description and their distribution across the training and testing sets.

Table 5.2: Aspect Categories Distribution in the Arabic Hotel Reviews Dataset.

Aspect Categories	Description	Dataset	
		Train	Test
Service	Opinions on staff attitude, such as room check-in, check-out, and reception.	1664	432
Hotel	Opinions evaluating the hotel as a whole.	1615	402
Food	Opinions on breakfast, food, drinks, specific dishes, and dining/drinking options.	775	195
Prices	Opinions on the prices of rooms, food & drinks, facilities, or the hotel in general.	486	100
Rooms	Opinions evaluating rooms based on condition, view, and furniture.	784	182
Facilities	Opinions on specific hotel facilities or guest services, including installations/areas (e.g., swimming pool, spa & sauna, beauty salon, restaurants, cafes).	892	226
Location	Opinions on the hotel's location, position, and surroundings.	890	249
Cleanliness	Opinions on the cleanliness and hygiene of bathrooms, common areas, and the hotel overall.	687	161
Amenities	Opinions focusing on the amenities included in rooms, such as air conditioning, toiletries, balcony, coffee maker, and linen.	443	102
Total Distribution		8236	2049

5.4.3 Aspect Sentiment

The Arabic hotel reviews dataset includes aspects that were classified as positive, negative, or neutral. For this project, we focused only on positive and negative sentiment classes and discarded the neutral class to simplify the sentiment analysis task. Table 5.3 displays how the three sentiment classes are distributed among the training and testing datasets.

Table 5.3: Aspect Sentiments Distibution in the Arabic Hotel Reviews Dataset.

Dataset	**Sentiment Classes**		**Total**
	Positive	**Negative**	
Train	5819	3141	8960
Test	1426	784	2210
Total	7245	3925	11170

5.5 Conclusion

In conclusion, choosing the Arabic hotel reviews dataset as an alternative proved to be a valuable decision. The dataset, with its large size, diverse aspect categories, and comprehensive annotations, is ideal for training and testing ABSA models. As a result of leveraging the hotel reviews dataset, challenges encountered during the annotation process were effectively resolved, contributing towards fulfilling the project's objectives and requirements.

CHAPTER 6

IMPLEMENTATION

In this chapter, we provide a comprehensive overview of the development process for Nire web-based tool that performs Aspect-Based Sentiment Analysis (ABSA) on public opinions of tweets about Saudi tourism. The chapter begins with a detailed system overview, followed by the activities and actions of the development and an outline of the tools and technologies utilized during the development process. Additionally, the chapter delves into the Artificial Intelligence (AI) components that drive the system, exploring their overall integration with Nire, models' implementation, and models' training. The chapter also provides detailed information regarding the front-end design, the back-end of Nire.

6.1 System Overview

Nire web-based system uses ABSA to analyze user-uploaded files. Once the user has uploaded a file, the system automatically conducts an ABSA on the textual contents of the file. It presents the results in an organized table, displaying the sentiments and aspects of each tweet in the file. Users can further access analysis through the dashboard, which presents the results in four plots. Simultaneously, Nire maintains the results in a history record that includes the name of the analysis, the date of the analysis, and the total number of tweets analyzed. Users can easily access and display the analysis dashboard again when needed.

6.2 Tools and Technologies

Nire system was developed using several tools. Python and its integrated development environments, Jupyter Notebook and Visual Studio Code, were employed for

coding and project presentation. Essential Python libraries included TensorFlow for deep learning models creation, Farasa for Arabic text pre-processing, and PyArabic for additional text handling utilities. The website's responsive front-end was designed using Bootstrap 5, while Flask served as the web framework due to its ease of use and scalability. Lastly, SQLite was selected as the project's lightweight and portable database management system. These tools collectively facilitated the creation of effective and efficient AI models and website.

6.3 Activities and Actions of the Development

The Nire web-based system consists of three layers. Each plays a role in delivering a seamless user experience. Our system layers are depicted in Figure 6.1, and they can be defined as follows:

1. **Front-end layer**: This layer includes a user interface for users to upload a file, view ABSA results on a dashboard, and other interfaces to interact with the system efficiently. In this layer, the user is required to follow these rules when uploading their file:
    - The file type must be either CSV or XLSX.
    - The file must have at least 5 rows of data.
    - The text data must be in the first column.
    - The text data must not exceed 256 characters.
2. **AI models layer**: It is the system's core layer, consisting of three components that work together to analyze data and efficiently produce ABSA results. The three components in this layer are:
    - Aspect Term Extraction.
    - Aspect Category Classification.
    - Aspect Sentiment Classification.
3. **Back-end layer**: This layer has two main components:
    - Database to store user information and ABSA results.

- Application programming interface to manage interactions between the front-end layer, AI models layer, and database.

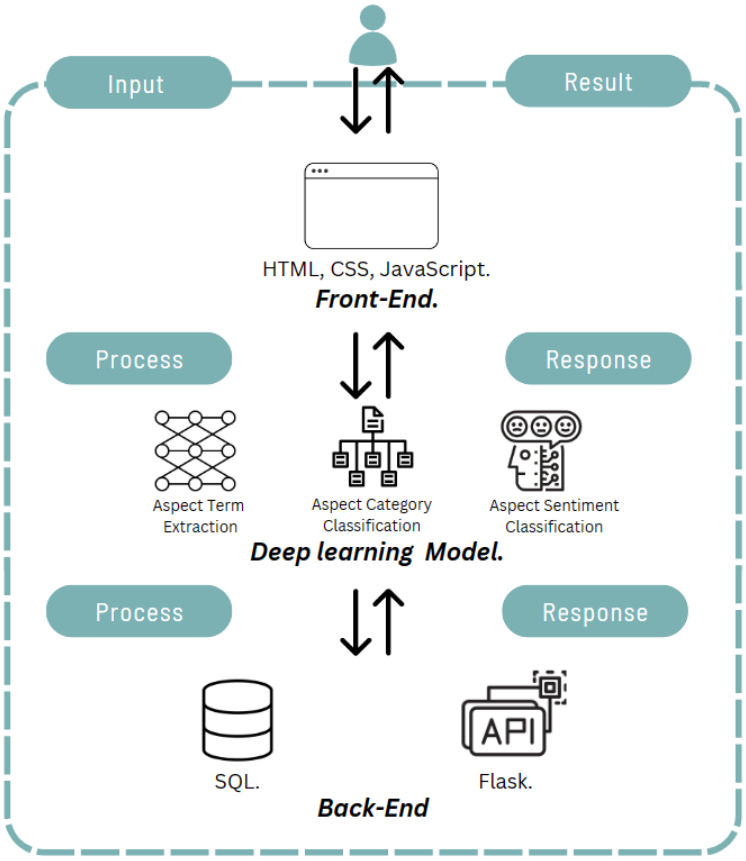

Figure 6.1: Nire System Framework.

6.4 AI Components Overview

The ABSA process comprises three primary AI components: aspect term extraction, aspect category classification, and aspect sentiment classification. Figure 6.2 presents an overview of these components integration and illustrates their interplay as a whole within Nire web-based system.

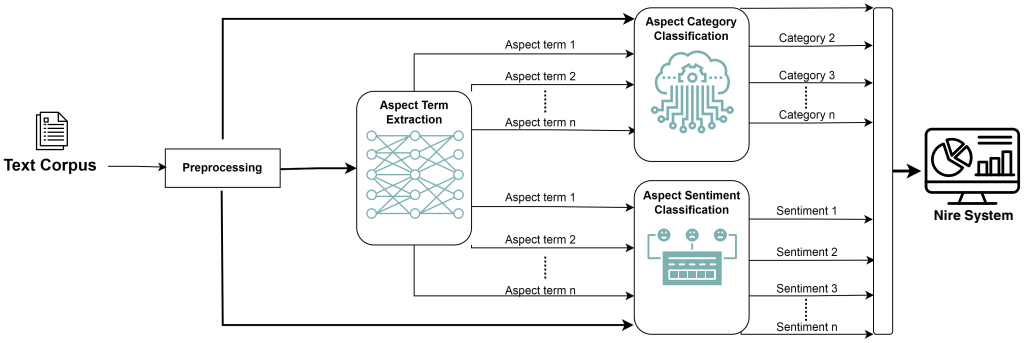

Figure 6.2: AI Components Integration Overview.

These components function together to analyze and classify aspects of input text, providing valuable insights for display on Nire's system. The implemented ABSA components operate as follows:

1. **Aspect Term Extraction**: This component identifies and extracts aspect terms in a sentence. Using the preprocessed text as an input, it outputs the extracted aspect terms.

2. **Aspect Category Classification**: This component, taking the extracted aspect terms and the preprocessed text as input, classifies each aspect term to one of the nine predefined categories. These categories include prices, service, food, hotel, cleanliness, rooms, amenities, facilities, and location. The output of this component is each aspect term, associated with its category.

3. **Aspect Sentiment Classification**: This component processes the aspect terms and the preprocessed text and effectively classifies the sentiment polarity (either positive or negative) for each term. The output of this step is a set of aspect terms each paired with its associated sentiment polarity.

While each component functions independently, they are closely intertwined in a sequential manner to accomplish the overall ABSA process. In the subsequent sections, we delve into the text preprocessing steps and the specific implementations of each component.

6.5 AI Models Implementation

In this section, we explore the implementation of our project's AI models, which correspond to the three components discussed above: aspect term extraction, aspect category classification, and aspect sentiment classification. Each component is implemented through a specific model, which is defined by its inputs, outputs, and architecture. We employed tailored preprocessing steps for Arabic text, preparing clean, consistent data. Finally, each model's architecture is built according to the task.

6.5.1 Pre-processing

In order to prepare the data for the models, it was necessary to preprocess the Arabic text. Several preprocessing steps were implemented in this project to ensure the input text is clean and consistent, making it easier for the models to understand and learn the patterns. The preprocessing steps used are summarized as follows:

- **Remove diacritics and elongation:**
  Our team removed diacritics to reduce the complexity of the text, as well as reducing the text to their standard forms to eliminate Tatweel (elongation). For example ممتـــــازٌ becomes ممتاز.

- **Remove punctuation marks and special characters:**
  Punctuation marks were removed to focus on meaningful content. Also, special symbols such as, &, _, $, were also removed.

- **Remove numbers and extra spacing:**
  As a general text cleaning process, the numbers were eliminated since they do not contribute to the aspect nor sentiment analysis. The extra spaces were also removed to maintain consistent formatting.

- **Remove mentions, hashtags, and hyperlinks:**

  When preprocessing twitter data, user mentions such as @Tourism, or hashtags like #tourist, are introduced. Our team removed these as well as hyperlinks to reduce the noise in the text data.

- **Remove emojis:**

  The developed emoticons icons are widely used on social media or the internet in general. However, they introduce noise and complexity in data and are always eliminated.

- **Stemming:**

  Using stemming, words were reduced to their roots, simplifying the vocabulary and helping the model identify patterns more easily by focusing on the core meanings of words rather than variations in their forms. For instance, الخدمات was reduced to خدمه.

6.5.2 Model 1: Aspect Term Extraction

In our system, Aspect Term Extraction (ATE) model is critical as it identifies terms expressing aspects in the hospitality domain. The overall performance and accuracy of Nire system depend on the successful extraction of aspect terms, as it directly impacts the effectiveness of other components, such as categorization and sentiment classification. This subsection focuses on the ATE model input features and labels.

As depicted in Figure 6.3, the ATE task is divided into two main parts. The first takes the input preprocessed text and tags it using the Begin-Inside-Outside (BIO) scheme.

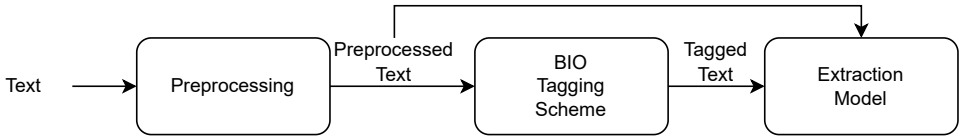

Figure 6.3: The Main Two Parts of the ATE Task.

This scheme is widely used in the literature for tasks such as named entity recognition and in our case, ATE. The BIO scheme assigns one of the three tags to each token in an input text: '**B**' for the beginning of an aspect term, '**I**' for inside an aspect term, and '**O**' for outside an aspect term. We further differentiated these tags by labeling them: B-ASP, I-ASP, and O. An example of an input text preprocessed and tokenized with each token assigned a corresponding tag is shown in Figure 6.4. With this automatic labeling process we were able to obtain the labels for our ATE model.

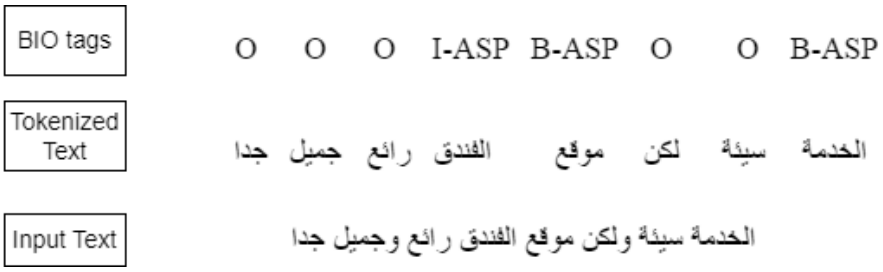

Figure 6.4: Implemented Labeling Scheme.

The second part of the task, extraction, is also depicted in Figure 6.3. This step involves the model using the preprocessed text as input features and the tagged text as input labels. These inputs are used when training the model itself, which is described in the upcoming sections.

6.5.3 Model 2: Aspect Category Classification

Aspect Category Classification (ACC) is the second model in our ABSA process. This model involves determining the specific category for the aspect term mentioned in the text. The ACC model uses the preprocessed text and the raw preprocssed aspect terms as input features. These features are then concatenated to form a unified representation of the text.

The model follows a multi-label classification approach. Which means that the concatenated input text can belong to one or more categories. As mentioned

in the previous section, nine categories in total serve as the input labels for the model.

6.5.4 Model 3: Aspect Sentiment Classification

The final model is Aspect Sentiment Classification (ASC). This model determines the sentiment polarity for each aspect term in the sentence. Our ASC model uses the preprocssed text and the raw preprocssed aspect terms as input features as well. It outputs either the positive or negative sentiment classes for each given aspect term.

6.5.5 Models Architectures

This subsection details the architectures of the AI models. While each model serves a specific purpose and has unique features, they all leverage the MARBERT embeddings for processing the Arabic language. We first introduce these shared embeddings before explaining the implementation of each model architecture.

6.5.5.1 MARBERT Embeddings

MARBERT embeddings form a critical foundation for all of our models, transforming the text into a numerical representation through contextual embeddings that can be processed by the models. In chapter 2, section 2.5.2, we introduced BERT (Bidirectional Encoder Representations from Transformers) as a powerful language model that generates context-aware word representations. MARBERT is a BERT variant trained on a vast corpus of Arabic text data (Abdul-Mageed et al., 2021). It forms the first layer in our models' architecture and functions as a feature extractor to produce contextualized word representations.

MARBERT embeddings blend three types of embeddings to deliver a comprehensive representation of the input data:

1. **Token Embeddings**: Represent each token in the input sequence. MARBERT's tokenizer converts every word into corresponding sub-word tokens.

2. **Positional Embeddings**: Capture the relative position of tokens within the input sequence, enabling the model to understand the relationship and order of words in a sequence.

3. **Segment Embeddings**: Distinguish between different input sequences, useful for tasks involving sentence pairs.

Following (Devlin et al., 2018), to leverage these embeddings, we first tokenized the input text and converted it into a format that accommodates the necessary embeddings. This process includes adding special tokens such as [CLS] and [SEP], which signal the beginning and separation of input sequences, respectively. As seen in Figure 6.5, the token, positional, and segment embeddings are then combined to yield a comprehensive representation of the input data, capturing not only the words but also their context. This rich representation forms the input features for our models.

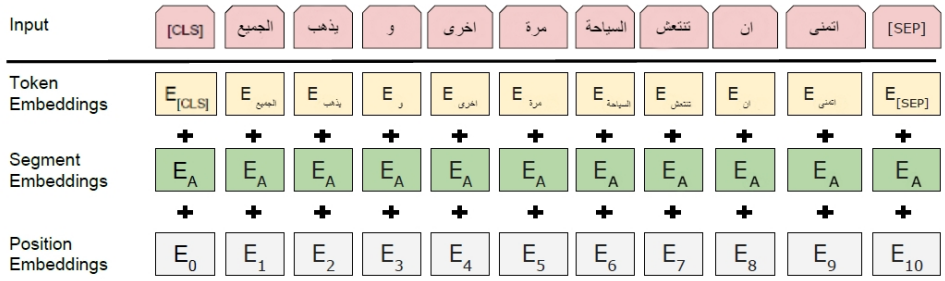

Figure 6.5: MARBERT Embedding Layers.
Source: (Devlin et al., 2018)

6.5.5.2 ATE Model Architecture

Given the importance of this component, we arrived at the final model architecture displayed in Figure 6.6, after conducting numerous experiments with different architectures. All the layers used were utilized heavily in literature, for instance, (Fadel et al., 2022) study. Regardless of the specific architecture under experimentation, all of them consistently incorporated the following layers:

1. MARBERT embeddings: Only required the token and positional embeddings.

2. Bidirectional Gated Recurrent Unit (BiGRU).

3. Output layer.

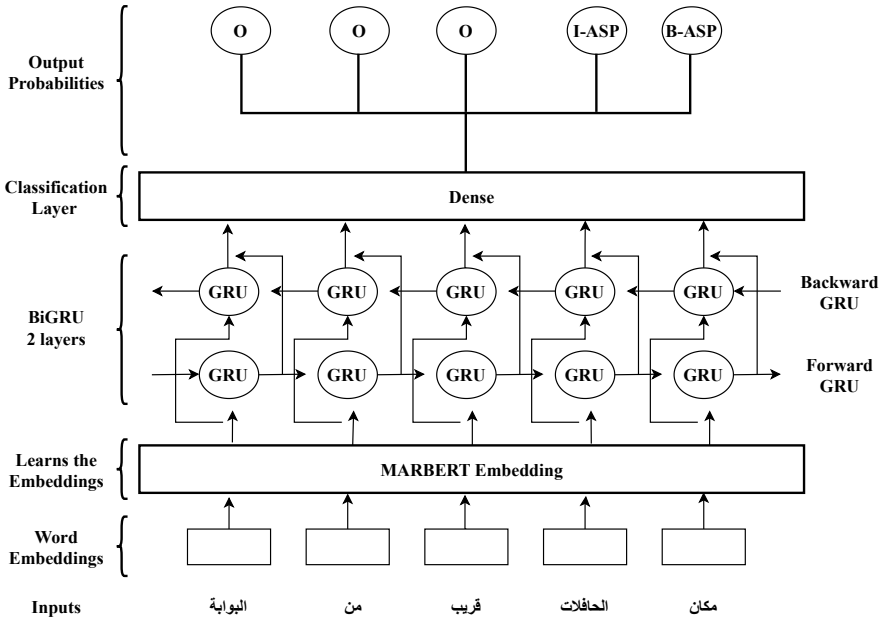

Figure 6.6: ATE Model Architecture.
Source: (Fadel et al., 2022)

However, each model variation is unique, with changes or additions made to these basic layers. The goal of these variations was to find the best structure for the ATE model.

In the upcoming section about model training, we provide an overview of each experimental architecture for the ATE model, and describe the arrangement of layers in each case while explaining the training process.

6.5.5.3 ACC Model Architecture

Our system's ACC architecture was developed following the model proposed by (Bensoltane and Zaki, 2021). Our team modified the mentioned study architecture by using MARBERT embedding as the first layer in our ACC component. Figure 6.7 illustrates the implemented model architecture.

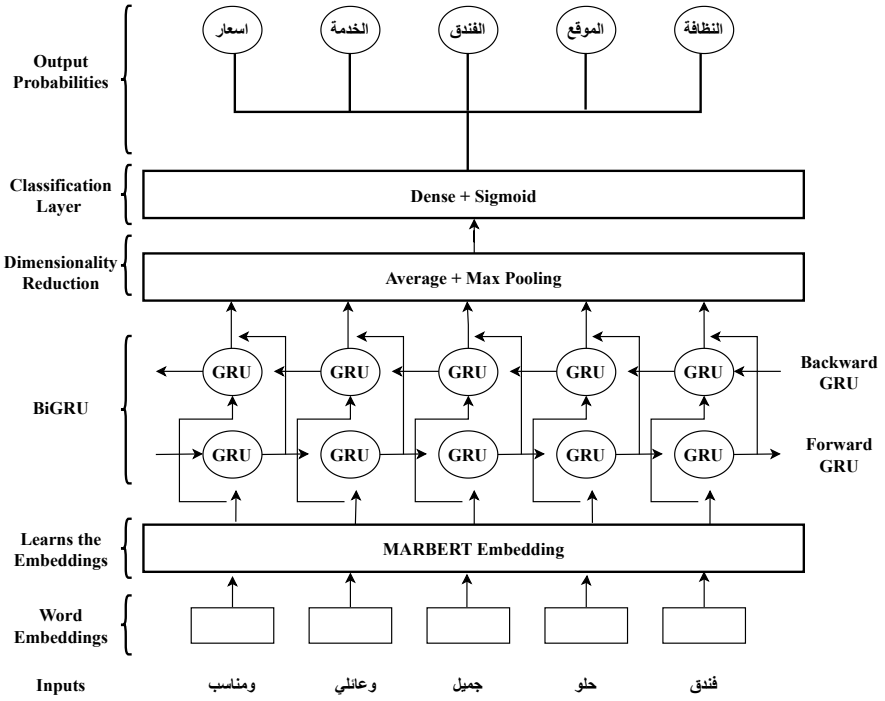

Figure 6.7: ACC Model Architecture.

The ACC architecture can be summarized in the following way:

1. **MARBERT embedding layer:** This layer generates contextualized vector representations of each word using the token embeddings and the positional embeddings.

2. **BiGRU layer:** This layer remembers long-term dependencies in both forward and backward directions.

3. **Dropout layer:** This layer regulates the model during training and helps avoid overfitting.

4. **Average pooling layer:** captures the context and general characteristics of the input text. In this way, it is possible to identify the relevant aspect categories based on the text's overall meaning.

5. **Max pooling layer:** This layer highlights the most prominent features in the input. It helps the model identify essential words or phrases indicative of a specific aspect category.

6. **Dense layer:** This layer, also known as the fully connected layer, outputs the final probabilities for each aspect category. In this layer, the combined

features from the previous layers are mapped to the nine possible aspect categories.

6.5.5.4 ASC Model Architecture

For our ASC model, it was developed as a MARBERT-based model. A MARBERT model utilizes deep neural networks, as well as bidirectional models based on transformers. Transformers are deep learning algorithms that rely on the attention mechanism to understand relationships among elements in a sequence (Vaswani et al., 2017). In Transformers, the attention mechanism enables the model to concentrate on specific elements of the input sequence when producing output. As a result, it assigns different importance scores according to the context for each element in the sequence (Vaswani et al., 2017). This approach allows the model to better capture long-range dependencies and complex relationships between words by paying attention to specific elements relevant to the current step in the process.

Figure 6.8 illustrates the overall architecture of the implemented ASC model as explained by (Abdelgwad et al., 2022a).

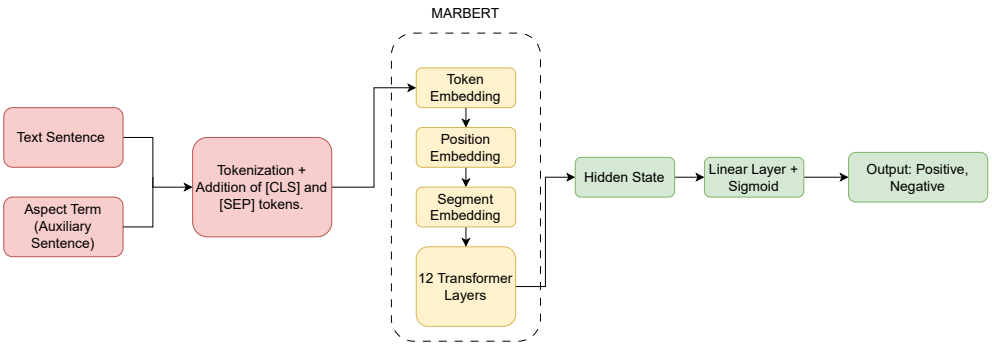

Figure 6.8: ASC Model Architecture.
Source: (Abdelgwad et al., 2022a)

- **Auxiliary sentence:**

  Due to the ability and effectiveness of the MARBERT model to deal with sentence pair classification tasks, and the fact that the MARBERT model can accept a single or pair of sentences as input, the ASC task was converted into a sentence-pair classification task utilizing the pre-trained MARBERT model, where the first sentence served as the input text, and the aspect terms served as an auxiliary sentence.

- **MARBERT segment embeddings:**

  The ASC model is similar to our two other components because it required the appropriate input representation to match MARBERT. Token and positional embeddings were incorporated into the structure the same way they were in our other models. However, additional embeddings called segment embeddings were required, for our sentence-pair classification model to distinguish between which tokens belong to which sentence.

6.6 AI Models Training

In this section, we discuss the training of our AI models using the Arabic hotel reviews dataset mentioned in Chapter 5. We begin by describing our data splitting approach, followed by an introduction to the evaluation metrics utilized to assess the performance of the models. Subsequently, we explain the training process for each specific model.

6.6.1 Dataset Split

The Arabic hotel reviews dataset was already pre-split into training and testing sets. We used the training set for model training, reserving 20% of it for validation purposes. This pre-existing split, along with the use of scikit-learn's train_test_split function, ensured balanced class distribution for each set.

6.6.2 Evaluation Metrics

In this project, we evaluated the models' performance using four widely used metrics:

1. **Accuracy**: Based on the number of predictions, the accuracy is calculated as a percentage. A higher accuracy indicates better performance.

2. **Recall**: A higher recall indicates better performance in identifying positive instances. It measures the proportion of true positives out of all positive instances.

3. **Precision**: Enhanced precision indicates a better ability to avoid false positives. The probability of true positives is computed based on the quantity of elements classified as positive.

4. **F1-score**: Taking into account both false positives and false negatives, F1-scores are used to measure precision and recall. A higher F1-score indicates higher overall performance. F1-score is calculated using the formula:

$$2 * (Precision * Recall)/(Precision + Recall) \tag{6.1}$$

6.6.3 ATE Model Training

Training our ATE model involved a series of experiments with different architectures. These experiments aimed to optimize the model's performance in extracting terms from Arabic text. The three main experiments we conducted each used a different model architecture, where we designed the architecture, trained the model using the training set, and evaluated its performance on a validation set to select the best architecture.

6.6.3.1 Experiment 1: MARBERT + BiGRU + CRF

In our first experiment, we extended the shared architecture mentioned in section 6.5.5.2 by adding a Conditional Random Field (CRF) layer to learn dependencies between the BIO-tagged labels, improving the overall token classification performance. This resulted in the following architecture: MARBERT Embeddings, BiGRU, and CRF. Figure 6.9 depicts the architecture of the first experiment.

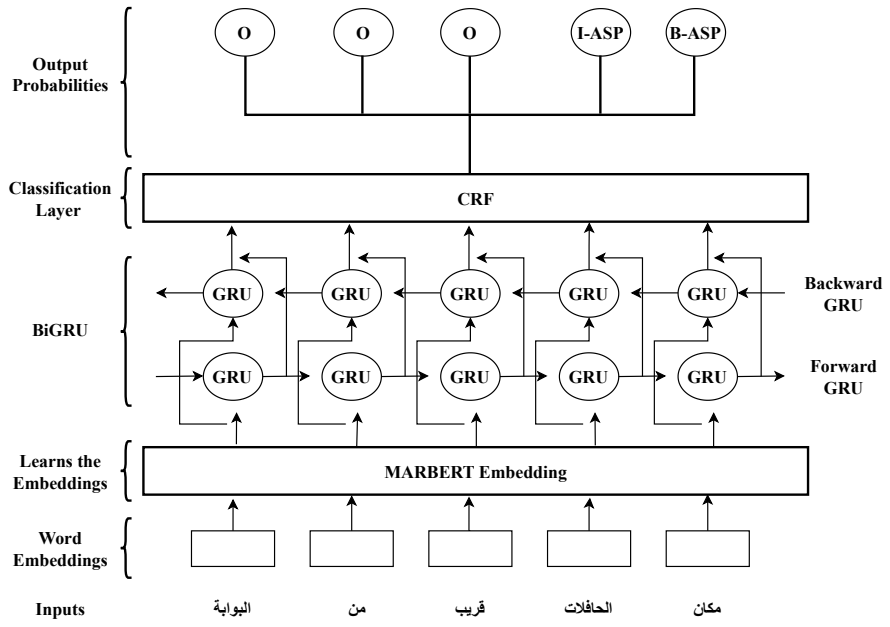

Figure 6.9: Experiment 1 Architecture.

To train this architecture, we began by splitting the dataset. The text was then preprocessed, tokenized, and tagged using the BIO scheme to obtain the input labels. Afterwards, we used a specialized tokenizer from the transformers library to tokenize the input preprocessed text data, ensuring compatibility with the MARBERT pre-trained model configuration. After initializing MARBERT's word embeddings with pre-set weights and introducing our prepared text, MARBERT converted these features into contextualized word embeddings. These contextualized embeddings were passed to the BiGRU layer, which enabled the model to identify complex patterns and relationships in the text. Lastly, the BiGRU's output was directed to the CRF layer, which used the BIO-tagged labels to generate predictions.

Working with this architecture, some challenges and limitations were encountered. The addition of the CRF layer increased the training time significantly, which posed a challenge for model development and optimization. Furthermore, there is no official implementation of the CRF layer in well-known deep learning frameworks like TensorFlow, which made this architecture more complex and less streamlined.

Despite the challenges posed by this architecture, we proceeded to optimize the model's hyperparameters during training using a suitable loss function and optimization algorithm. Our hyperparameters setting is described in Table 6.1. To ensure the model's applicability to new data, we monitored its performance on the validation set throughout the training process. After completing the training, we evaluated the model's effectiveness for ATE by assessing its performance. The model achieved a precision of 83%, recall of 57%, and F1-score of 64% on the validation data. Although these results were satisfactory for the ATE task, our experience with the challenges and limitations of this architecture motivated us to explore alternative architectures in our subsequent experiments.

Table 6.1: Experiment 1 Hyperparameters Setting.

Parameter	**Value**
Learning Rate	1e-5
Dimension Dim	768
Batch Size	32
Dropout	0.2
Hidden GRU States	128
Optimizer	Adam
Epochs	30

6.6.3.2 Experiment 2: MARBERT + Stacked BiGRU

In this experiment, we modified the shared architecture by stacking two layers of BiGRU and adding a dense layer instead of the CRF layer in the previous experiment. This resulted in the following architecture: MARBERT Embeddings, BiGRU (2 layers), as seen in figure 6.10.

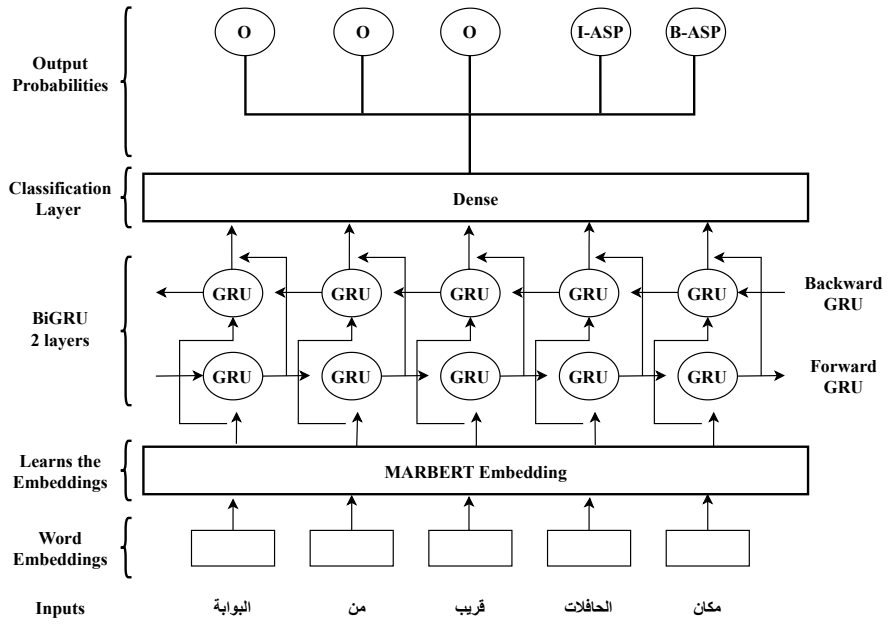

Figure 6.10: Experiment 2 Architecture.

The training followed in this experiment is similar to experiment 1, except that the contextualized embeddings obtained from MARBERT were passed to an additional BiGRU layer. Then, a final dense layer output the predicted probabilities for each tag. According to (Fadel et al., 2022), when stacking BiGRU layers,

the performance of their proposed architecture has improved, which motivated our team to try a similar approach. In an equal manner to experiment 1, we evaluated this architecture with different hyperparameters settings and utilized the setting in Table 6.2 for each parameter.

This experiment exhibited overfitting issues in the training process, which is common in deep learning models where the model learns the training data too well, capturing noise rather than the underlying pattern, and thus performs poorly on unseen data. Our team addressed this issue by increasing the dropout rate and performed early stopping on the validation set. Early stopping monitored the model performance on the validation set and stopped the training when the model was not performing well. Next, the model was evaluated on the validation set, achieving a precision of 81%, recall of 73%, and F1-score of 77%.

Table 6.2: Experiment 2 Hyperparameters Setting.

Parameter	**Value**
Learning Rate	2e-5
Dimension Dim	768
Batch Size	32
Dropout	0.5
Hidden GRU States	256
Optimizer	Adam
Epochs	30

6.6.3.3 Experiment 3: MARBERT + BiGRU

In this final experiment, we used the common architecture utilizing only one layer of BiGRU and adding a dense layer. The arcitecture is illustrated in figure 6.11.

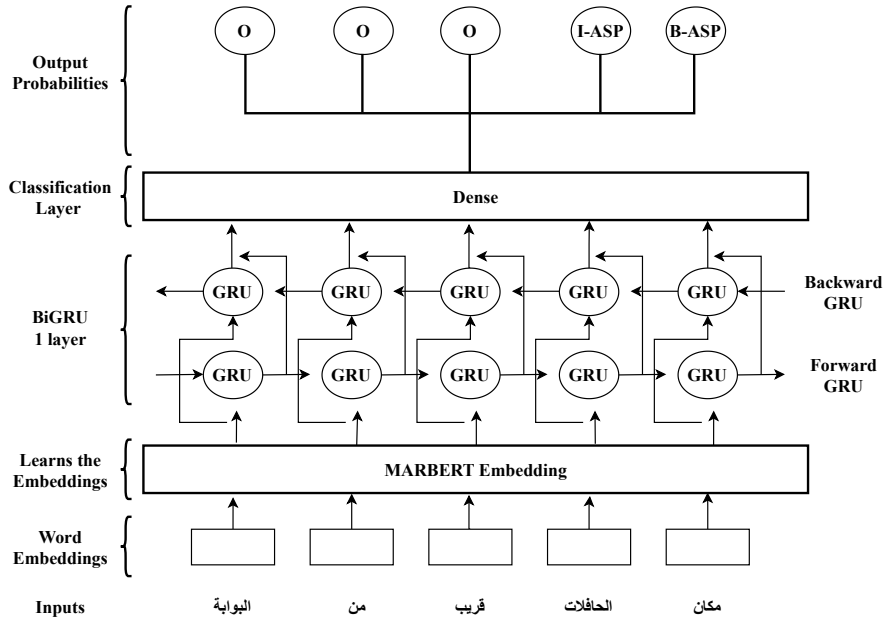

Figure 6.11: Experiment 3 Architecture.

As for training, the contextualized embeddings received from MARBERT were passed to the BiGRU layer. Following that, the dense output layer was responsible for predicting the labels. The same hyperparameters setting was used as in experiment 2. This architecture was considered in order to determine what would happen if our model adopted a more simple architecture with fewer layers. When evaluated it on the validation set, this architecture delivered a precision of 80%, recall of 61%, and F1-score of 66%.

6.6.3.4 ATE Model Selection

As explained in the previous experiments, we evaluated the performance of our architectures on the validation set using standard evaluation metrics, such as precision, recall, and F1-score. The architectures' performance on the validation set was used to compare them and select the most efficient one for the final ATE model.

As shown in Table 6.3, experiment 2 performed the best among the other two. Experiment 1 had some limitations and challenges, which led us not to explore any improvement techniques for it. Instead, we focused on enhancing the

MARBERT + Stacked BiGRU model from experiment 2, which demonstrated the best performance while avoiding the limitations encountered in other experiments.

Table 6.3: ATE Model Experiments.

Experiment Type	Precision	Recall	F1-score
Experiment 1	83%	57%	64%
Experiment 2	**81%**	**73%**	**77%**
Experiment 3	80%	61%	69%

6.6.4 ACC Model Training

The first step in training our ACC model was to preprocess and tokenize the input text. This involved generating contextualized vector representations for each word using the MARBERT embedding layer. Furthermore, we used the raw aspect terms that are labeled in the Arabic hotel reviews dataset as additional features for this model. The aspect terms were incorporated into the model by concatenating their embeddings with the input text embeddings.

Once the input features were prepared, we moved on to encoding the aspect category labels. We used an appropriate encoding scheme called label binarization, which converted the categorical labels into a binary matrix representation. This binary representation enabled our model to handle multi-label classification tasks, where each input can be assigned multiple aspect categories.

The embeddings passed through several layers, including a BiGRU, Dropout, Average Pooling, and Max Pooling. The model learned to identify relevant patterns and notable features in the input through these layers. The features extracted from the input were then associated with the encoded aspect category labels in the Dense output layer.

We used the Adam optimization function and the binary cross-entropy loss function to train our ACC model. The model underwent training over ten trials. Table 6.4 summarizes the hyperparameters used in this model. The ACC model achieved a precision of 93%, a recall of 78%, and an F1-score of 84% on the training set.

Table 6.4: ACC Model Hyperparameters Setting.

Parameter	**Value**
Learning Rate	1e-3
Dimension Dim	768
Batch Size	32
Dropout	0.25
Hidden GRU States	128
Optimizer	Adam
Loss	Binary-cross entropy
Epochs	10

6.6.5 ASC Model Training

In this final model, our first step was to preprocess the text data and raw aspect terms labeled in the Arabic hotel reviews dataset. Following that, we tokenized the sentences paired with the aspect terms, and prepared the input according to MARBERT's requirements.

We used transfer learning to fine-tune the model for our task. The model was initialized with the pre-trained weights and continued training on our dataset, adjusting the model's parameters to fit the ASC task better. A dropout layer was added with a 25% rate to regulate the model, and early stopping was also utilized to monitor the validation set.

Our model also included a linear fully connected output layer to classify sentiment. The output layer took the last hidden state from MARBERT and used it for classification.

The Adam optimizer was used during training with a 1e-5 learning rate, the rest of the hyperparameters setting is shown in Table 6.5. The model was trained for four trials to achieve the optimal performance. The model yielded accuracy of 95% on the training set.

Table 6.5: ASC Model Hyperparameters Setting.

Parameter	**Value**
Learning Rate	1e-5
Batch Size	16
Dropout	0.25
Optimizer	Adam
Loss	Binary-cross entropy
Epochs	4

6.7 Web System Implementation

6.7.1 Front-end Implementation

The front-end of Nire website was developed using Bootstrap 5, a widely-used front-end framework that enables the creation of responsive websites using HTML, CSS, and JavaScript. This section presents an overview of the front-end interfaces of the Nire website, depicted in Figure 6.12, followed by a concise explanation of each interface. Appendix III contains snapshots of some of Nire implemented interfaces.

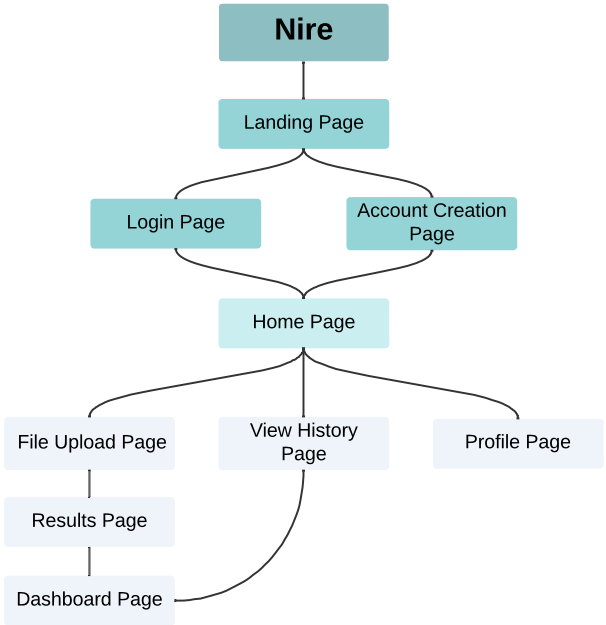

Figure 6.12: Web-Based Map of Nire.

- **Landing page:** The landing page on the Nire website introduces our team, goals, and features of the Nire tool. It serves as a gateway and initial connection point for visitors to access and utilize the tool on the main website.

- **Login page:** Users must input their email addresses and passwords to log in to the platform. If there are errors in the password or email entered, the system displays an error message.

- **Account Creation page:** The Nire website's account creation page requires users to input their first name, last name, email address, password, and password confirmation. These details are essential for successful account creation. The system includes error handling to prompt error messages for incorrect password input or duplicate email addresses.

- **File Upload page:** The file upload page allows users to upload tweets for ABSA. Only CSV or Excel file formats are accepted for upload. If an incorrect file format is uploaded, the system promptly displays an error message, guiding users to submit the correct format.

- **Result page:** The results page presents a table displaying tweets, their corresponding aspects, and sentiments. This tabular format provides a clear overview of the analyzed data. Additionally, users can navigate to the dashboard for a more analytical view of the results.

- **Dashboard page:** The dashboard page presents ABSA outcomes in four visual formats, including a pie chart, a bar chart, a radar chart, and a word cloud. These dynamic and engaging visualizations offer insights into the analyzed data in a user-friendly manner.

- **Profile page:** The profile page displays the user's information, including first name, last name, email address, and password. Users can edit their information and make updates as needed. Additionally, the profile page allows users to delete their accounts if desired.

- **View History page:** The view history page displays a table containing each analysis date, name, and the number of associated tweets. Users can easily access the resulting dashboard to view the outcomes of their past analyses.

6.7.2 Back-end Implementation

For the backend of Nire, we selected Flask as our web framework. Its lightweight and flexible toolset enabled us to swiftly define routes and functions, interact with our database using SQLalchemy, and render templates to create dynamic web pages. SQLite, a lightweight database, was employed to manage our web application's data and business logic with high efficiency. Through this process, Flask proved to be a powerful and versatile tool, playing a pivotal role in managing the backend of Nire effectively.

6.8 Conclusion

In this chapter, we described the development process of creating Nire, a web-based ABSA tool for analyzing public opinions of Saudi tweets. The chapter encompasses various aspects such as system overview, tools and technologies used, AI models integration and implementation, front-end implementation, back-end creation.

CHAPTER 7

TESTING

The system development process involves a crucial phase known as system testing, which plays a vital role in validating the performance and functionality of the system. It is imperative to thoroughly examine the system for flaws or vulnerabilities across all its elements. This stage involves testing the system to make sure that each of its parts is operating as intended. Additionally, the testing procedure determines whether the system meets its specified requirements. This chapter explores various testing techniques and provides the AI models testing. Subsequently, the chapter delves into unit testing, encompassing the back-end unit testing. Lastly, we conduct usability testing using quantitative and qualitative measures.

7.1 AI Models Testing

Our AI models for Aspect-Based Sentiment Analysis (ABSA) consist of three models, each addressing a specific task: aspect term extraction, aspect category classification, and aspect sentiment classification. Each model was tested individually using the test set from the pre-split Arabic hotel reviews dataset explained thoroughly in Chapter 5. In the following subsections, we describe each model's testing.

7.1.1 Aspect Term Extraction Testing

The Aspect Term Extraction (ATE) model was trained on input raw text and corresponding tagged text as input labels. Each token in the tagged text was annotated according to the Begin-Inside-Outside (BIO) tagging scheme, marking

them as one of three classes: 'O', 'B-ASP', and 'I-ASP'.

Notably, the model was trained on stemmed preprocessed text, and thus extracts stemmed aspect terms. This process can increase the model's efficiency by reducing the feature space, but it might also affect the readability of the extracted aspects and potentially lose some nuances in meaning, as stemming reduces words to their root form.

Upon being trained with these inputs, the ATE model can process unseen text by predicting one of these three classes for each token in the input. It is essential to recognize that there is a class imbalance, as 'O' tags, indicating non-aspect terms, are the most frequent. Therefore, this imbalance should be considered when interpreting the model's performance.

As shown in Figure 7.1, the model demonstrates high performance for the O class with an F1-score of 99%, confirming its effectiveness in distinguishing non-aspect tokens. However, the F1-scores for B-ASP and I-ASP are 72% and 48%, respectively, highlighting areas of challenge, particularly for the I-ASP class.

	precision	recall	f1-score	support
O	0.99	1.00	0.99	154097
B-ASP	0.75	0.69	0.72	2253
I-ASP	0.61	0.40	0.48	706
accuracy			0.99	157056
macro avg	0.78	0.69	0.73	157056
weighted avg	0.99	0.99	0.99	157056

Figure 7.1: Classification Report for the ATE Model on the Test Set.

The lower performance on the 'I-ASP' class stems from the difficulty in detecting multi-word aspect terms and the fewer examples of such terms in the training set. Figure 7.2 further demonstrates the model's performance by presenting its aspect term predictions for a sample input.

To conclude, while these scores reflect a promising start, they also emphasize the challenges of ATE, particularly in detecting multi-word aspect terms. These insights help inform future work and improvements for this model.

```
# Example usage
marbert_preprocessor=MarbertPreprocessor(max_seq_len=128)
text= "المنطقة في غاية النظافة وقسم الاكل ممتاز وفيه جميع المطاعم المعروفة"
#pass the text, preprocessor and the ATE model to the function to predict.
aspects = extract_aspects(text, marbert_preprocessor, model2)
print("Input Sentence:" , text)
print("Extracted aspects:", aspects)
```

✓ 0.8s

```
1/1 [==============================] - 0s 64ms/step
Input Sentence: المنطقة في غاية النظافة وقسم الاكل ممتاز وفيه جميع المطاعم المعروفة
Extracted aspects: ['مطعم', 'قسم اكل', 'منطقة']
```

Figure 7.2: Sample Sentence Given to the ATE Model for Prediction.

7.1.2 Aspect Category Classification Testing

Aspect Category Classification (ACC) is used to categorize aspect terms. As a result of training the model, it can classify an input text with raw aspect terms as input as well into one of nine category classes. Figure 7.3 illustrates the model's performance on the test set, showing its ability to accurately assign categories.

	precision	recall	f1-score	support
Prices	0.81	0.74	0.77	100
Service	0.97	0.86	0.91	432
Food	0.92	0.80	0.86	195
Rooms	0.74	0.71	0.73	182
Hotel	0.89	0.52	0.66	402
Facilities	0.71	0.65	0.68	226
Location	0.88	0.71	0.78	249
Cleanliness	0.75	0.83	0.78	161
Amenities	0.91	0.38	0.54	102
micro avg	0.85	0.70	0.77	2049
macro avg	0.84	0.69	0.75	2049
weighted avg	0.86	0.70	0.77	2049
samples avg	0.84	0.75	0.77	2049

Figure 7.3: Classification Report for the ACC Model on the Test Set.

The model exhibits a high performance in the 'Service' category, with an F1-score of 91%. This result indicates the model's strong capability in identifying and categorizing aspect terms related to 'Service'. However, the model's performance in the 'Amenities' category shows room for improvement with an F1-score of 54%.

Figure 7.4 further illustrates the model's performance by providing an example of a sentence used to predict the aspect category of each given aspect term.

```
# Example input text and aspect terms given to the model
text = "المنطقة في غاية النظافة قسم الاكل ممتاز وفيه جميع المطاعم المعروفة"
aspects = ["منطقه", " قسم الاكل", "مطعم"]
#pass the text, aspects and our model
predicted_categories = detect_category(text, aspects, model, tokenizer, label_names)
# display the predicted category of each aspect term
for aspect, category in predicted_categories.items():
    print(f"Aspect term: {aspect} Category: {label_mapping[category]}")
```

✓ 0.1s

```
1/1 [==============================] - 0s 66ms/step
Aspect term: منطقه Category: Cleanliness
Aspect term:  قسم الاكل Category: Food
Aspect term: مطعم Category: Facilities
```

Figure 7.4: Sample Sentence Given to the ACC Model for Prediction.

7.1.3 Aspect Sentiment Classification Testing

The Aspect Sentiment Classification (ASC) model was tested by providing input text and raw aspect terms. The model was then tasked with predicting one of two classes: positive or negative. Testing was performed on the test set, and the classification report demonstrated the model's overall performance, achieving a high accuracy of 95% as depicted in Figure 7.5.

```
70/70 [==============================] - 17s 167ms/step
              precision    recall  f1-score   support

    negative       0.93      0.92      0.92       784
    positive       0.95      0.96      0.96      1426

    accuracy                           0.95      2210
   macro avg       0.94      0.94      0.94      2210
weighted avg       0.95      0.95      0.95      2210
```

Figure 7.5: Classification Report for the ASC Model on the Test Set.

Additionally, the F1-score is 94%, which confirms the model's robustness. Analyzing this measure for each class, the model performed slightly better in recognizing positive sentiment with an F1-score of 96%, while negative sentiment yielded an F1-score of 92%. Despite this slight discrepancy, both classes performed well.

A sample input was also used to illustrate the outcome of the model's sentiment prediction task as seen in Figure 7.6.

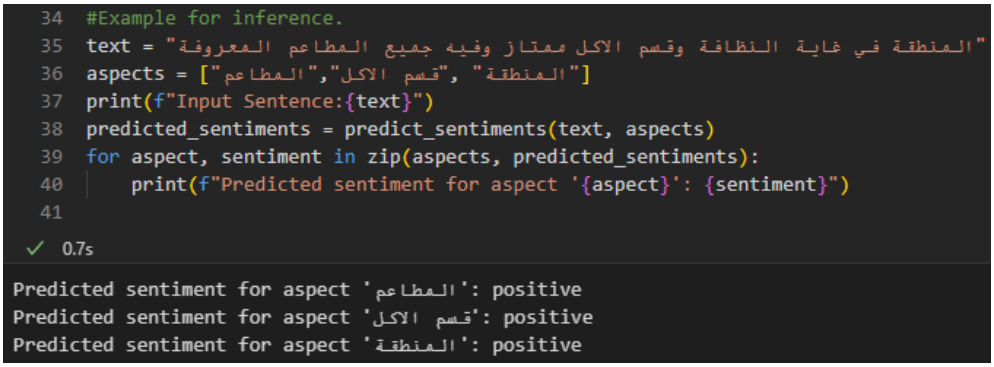

Figure 7.6: Sample Sentence Given to the ASC Model for Prediction.

7.2 System Testing

System testing is an important phase in system development that helps identify issues before deployment. It involves a series of tests that evaluate the system's functionality and the interactions between its various components. Nire system is composed of three primary layers: the user interface, the AI models, and the backend. A smooth functioning of the system depends on each of these layers. Testing ensures seamless data flow between the layers, accurate data presentation to end-users, and correct data retrieval from the database. The two primary testing techniques employed for Nire system include:

- Unit Testing.
- Usability Testing.

Unit Testing verifies the internal workings of system components and functions, ensuring they operate as intended. On the other hand, usability testing evaluates the system by engaging targeted users in objective and subjective evaluations.

7.3 Unit Testing

Unit testing is a crucial part of the software development process as it ensures the correct functioning of individual software units. Its significance lies in the thorough examination of individual components to ensure their proper operation.

In the case of our Nire system, we specifically focused on testing the backend components to guarantee their independent functionality and reliability. The next subsection presents the backend unit testing conducted within our Nire system. Given the integration of the AI models, a comprehensive testing phase was undertaken to evaluate the performance of these components after integration. The results obtained from the test is summarized in Table 7.4, providing an overview of the outcomes achieved for each specific task.

7.3.1 Backend Unit Testing

The backend unit testing contained manual testing of each component via the GUI to guarantee the correct functioning of backend elements. Additionally, comprehensive database unit testing was conducted to validate its functionality. It explicitly focused on the insert, retrieval, and deletion operations. Tasks conducted for the database unit testing are outlined in Table 7.1 with the results.

Table 7.1: Database Unit Testing Tasks

Database Unit Test	Purpose	Pass/Fail
Store User Information	Verify correct insertion of user information into the User table.	Pass.
Update User Information	Ensure proper modification of user information in the User table.	Pass.
Store Analysis Results	Confirm accurate storage of analysis results in the Aspect and Sentiment table.	Pass.
Store Analysis Information	Verify proper storage of analysis details in the Aspect Sentiment Analysis and History table.	Pass.
Delete Account	Ensure correct removal of user account information from the User table.	Pass.
View Results of the Analysis	Validate the retrieval of analysis results from the Aspect and Sentiment table.	Pass.
View User Information	Confirm the retrieval of user information from the User table.	Pass.
View Previous Analysis Operations	Ensure the successful retrieval of previous analysis operations from the Aspect Sentiment Analysis and History table.	Pass.

7.4 Usability Testing

The usability of Nire's website has been evaluated by its target end-users. This process involved observers carefully monitoring and taking notes while participants complete specific tasks. The primary aim of usability testing is to measure the system's ease of use, gather both qualitative and quantitative data, and determine participant satisfaction levels.

7.4.1 Test Participants

The participants were different individuals, some with a keen interest in data analysis and opinion mining, and others were students in computer science.

7.4.2 Environment of the Test

Testing was carried out remotely from participants' homes and workplaces. Some settings were noisy, while others were quiet, enabling a realistic evaluation of the Nire system in a variety of real-life settings.

7.4.3 Evaluation Tasks

In the conducted usability testing of Nire, the following tasks were undertaken to evaluate the user experience of the system. Each task represented a specific action or interaction that users performed while using Nire. The objective was to assess the system's ease of use, efficiency, and effectiveness in accomplishing these tasks. The completed tasks were as follows:

1. Create account: The user was asked to create a new account by correctly filling in their details in the registration form, then clicking the "Sign Up" button.

2. Log in: The user was asked to log in to the website by entering their correct email and password, then clicking the "Log In" button.

3. Upload file: The user was provided with CSV/XLSX files. The user was asked to input a name for their analysis, and upload a file.

4. View results: Upon completion of the analysis, the user was redirected to the results page, where they reviewed the findings.

5. Display dashboard: The user was asked to navigate to the dashboard to view the analysis in plots and charts.

6. View history records: The user was asked to access the history records page to review a list of their previous analyses.

7. Display dashboard from a history record: The user was asked to select a specific analysis from the history records and click the "View" button to display the associated dashboard.

8. Edit account information: The user was asked to navigate to the profile page, update their account details as necessary, and click the "Save

Changes" button.

9. Delete account: For users who wished to test the account deletion process, they navigated to the profile page, clicked the "Delete Account" button, and confirmed their decision.

10. Log out: If the user wished to end their session, they were asked to navigate to the website's header and click the "Log out" button.

7.4.4 Objectives Measure Analysis

In the performed usability testing, the objective measurements implemented serve as the standard for comparing user interactions with the system. These benchmarks include two measures: the number of clicks made during the execution of tasks and the duration required to complete each task. Each following point discusses these measures and compares them against the expected values of each task. Appendix IV presents all participants' usability testing results in detail.

- **Number of Clicks:** Our test evaluated the simplicity and intuitiveness of the system by recording the number of clicks made by the 7 participants during the execution of the tasks. We then compared these actual values to an expected standard number of clicks, established based on expert analysis. The data, categorized according to participants' minimum, maximum, and average number of clicks, are presented in Table 7.2. Notably, our analysis demonstrated that all tasks yielded acceptable results in terms of the number of clicks, aligning well with the expected values, thereby indicating a positive user interaction with the system.

- **Task Completion Duration:** As shown in Table 7.3, participants' completion times aligned well with expectations. However, Task 4, involving the analysis of an uploaded file and subsequent review of results, demonstrated a significant variation due to the impact of the content size (amount of text) within the file on processing time. Participants' duration for this task ranged from 56 seconds to 3 minutes, which falls within the expected time frame of 39 seconds to 3 minutes. While the ABSA models effectively

Table 7.2: Participants' and Expected Number of Clicks.

Task	Participants Number of Clicks			Expected Number of Clicks
	Min Number of Clicks	Max Number of Clicks	Avg Number of Clicks	
1	2	4	2	2
2	3	3	4	3
3	5	7	6	5
4	10	15	12	12
5	11	16	13	13
6	1	2	1	1
7	2	6	3	2
8	2	3	2	2
9	3	5	4	3
10	1	1	1	1

processed files with smaller amounts of text, they exhibited extended processing times for files with more textual data. Despite the task duration falling within expectations, there is still room for improvement. Future enhancements should increase system efficiency and performance, particularly when handling large text data files.

Table 7.3: Participants' and Expected Duration.

Task	Participants Amount of Duration			Expected Amount of Duration
	Min Amount of Duration	Max Amount of Duration	Avg Amount of Duration	
1	50s	2m	1m, 10s	1m, 30s
2	16s	1m, 30s	32s	20s
3	20s	37s	28s	20s
4	56s	3m	1m, 7s	39s - 3m
5	10s	25s	17s	12s
6	7s	31s	14s	10s
7	26s	1m, 12s	26s	10s
8	14s	33s	23s	16s
9	8s	35s	14s	10s
10	3s	10s	6s	5s

7.4.5 Subjective Measure Analysis

Subjective measure analysis focused on users' opinions about their experiences. This project used a post-test survey to assess participants' satisfaction and ease of navigation on our website. Additionally, we included an open-ended question,

which provided valuable insights into user perspectives and identified opportunities for enhancing the user experience.

1. Ease of Navigation

   The ease of navigation on a website is an essential element of the user experience. Our analysis aimed to assess participants' views on our website's navigability. As seen in Figure 7.7, the results of our analysis showed that 85.7% of participants found the website easy to navigate.

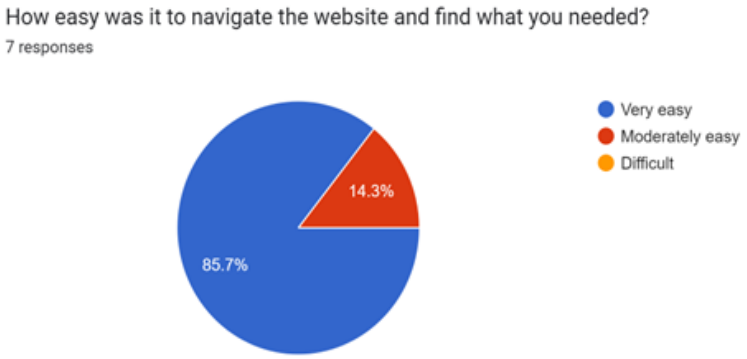

Figure 7.7: Results of How Easy the System was to Navigate for Participants.

2. Satisfaction

   A key metric of a website's success is user satisfaction. We assessed participants' satisfaction levels to evaluate their overall experiences with our website. Based on the obtained results, 57.1% were satisfied, while 42.9% were moderately satisfied. The analysis is summarized in Figure 7.8.

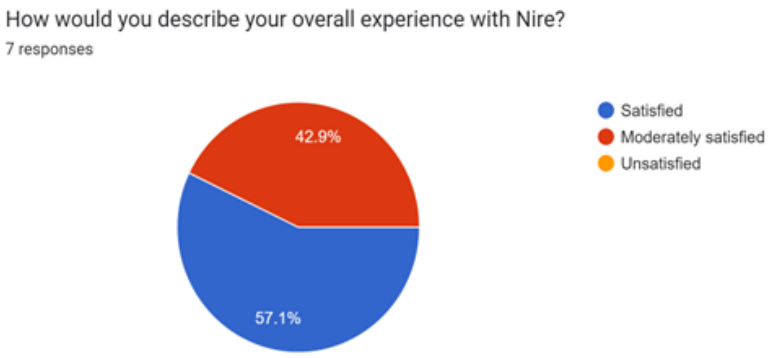

Figure 7.8: Results of the Overall Satisfaction of the Participants.

3. Feedback

   Open-ended feedback from users offers insights into their personal experiences. We used this feedback to understand user perspectives and identify areas for potential improvement. Some of the feedback is listed as follows:

   - Addition of a help/documentation page.
   - Implement a feature for deleting unnecessary history records.

7.5 Conclusion

In conclusion, this chapter highlighted the vital role of system testing in validating the performance, functionality, usability of the Nire system. The system was thoroughly examined to ensure flawless operation and compliance with specified requirements. Through the exploration of various testing techniques, the establishment of required software testing methods, and the execution of AI models testing, unit testing, and usability testing, the system's performance and usability were successfully confirmed.

Table 7.4: Nire System Testing

Test Name	Test Description	End-User Input	Expected output	Actual output	Pass/Fail
Create Account	Verify successful user registration with valid details.	First Name, Last Name, Email, Password, Repeated Password.	User should be successfully registered.	The output was the same as the expected output.	**Pass**
Create Account - With existing email.	Verify that an error message is displayed for an existing email.	First Name, Last Name, Already Existing Email, Password, Repeated Password.	Error message should be displayed, indicating that the email already exists.	The output was the same as the expected output.	**Pass**
Log In	Verify successful user login with valid information.	Email, Password	User should be successfully logged in.	The output was the same as the expected output.	**Pass**

Continued on next page

Table 7.4 Nire System Testing (Continued)

Log In - With non-existent Email.	Verify error message for non-existent email during login.	Non-existent Email, Password	Error indicates that email not found in database.	The output was the same as the expected output.	**Pass**
Log In - With incorrect password.	Verify that an error message is displayed for an incorrect password.	Email, Incorrect Password	An error message should be displayed, indicating an incorrect password.	The output was the same as the expected output.	**Pass**
Upload File	Verify successful file upload.	Analysis name, File	File should be successfully uploaded and sent for analysis.	The output was the same as the expected output.	**Pass**
Upload File - Empty Analysis Name.	Verify error handling for empty analysis name during uploading file.	Analysis Name	Error message should be displayed, indicating a required field is missing.	The output was the same as the expected output.	**Pass**

Continued on next page

Table 7.4 Nire System Testing (Continued)

Upload File - Invalid File Format.	Verify error handling for invalid file formats during file upload.	Analysis name, Unsupported file	Error message should be displayed, indicating unsupported format.	The output was the same as the expected output.	**Pass**
View Result	Verify display of analysis results after upload file.	N/A	Analysis results should be successfully displayed.	The output was the same as the expected output.	**Pass**
View Result by Sentiment	Verify display of analysis results by sentiment (positive/negative).	N/A	Analysis results of the chosen sentiment should only be displayed.	The output was the same as the expected output.	**Pass**
Display Dashboard	Verify display of analysis results in dashboard.	N/A	Analysis results should be successfully represented in dashboard.	The output was the same as the expected output.	**Pass**

Continued on next page

Table 7.4 Nire System Testing (Continued)

View Account Information	Verify the functionality to view user profile.	N/A	User account information should be successfully displayed.	The output was the same as the expected output.	**Pass**
Edit Account Information	Verify user information editing functionality.	First Name, Last Name, Email, Password, Repeated Password.	User account information should be successfully updated.	The output was the same as the expected output.	**Pass**
Delete Account	Verify account deletion functionality.	N/A	User account should be successfully deleted.	The output was the same as the expected output.	**Pass**
View History	Verify display of user's past operations.	N/A	User's past operations should be successfully displayed.	The output was the same as the expected output.	**Pass**

Continued on next page

Table 7.4 Nire System Testing (Continued)

Display Dashboard from a History Record	Verify the functionality to displayed dashboard from the history records.	N/A	The dashboard of the selected record should be displayed.	The output was the same as the expected output.	**Pass**
Log Out	Verify the functionality to log out of the system.	N/A	User should be successfully logged out.	The output was the same as the expected output.	**Pass**

CHAPTER 8

CONCLUSION AND FUTURE WORK

Nire's development journey has been documented in the preceding chapters, following the iterative waterfall methodology. Each project stage has been thoroughly explored. This report showcases the project's evolution, starting with problem definition and project details, followed by research, requirement gathering, system design, implementation, and final testing. By leveraging the team's collective expertise and knowledge, the project integrates ideas from diverse disciplines, effectively bridging theoretical knowledge with practical application. This chapter focuses on the challenges encountered during the project's development, the lessons learned, and the skills acquired. Additionally, insights into future work for Nire are provided.

8.1 Challenges and Difficulties

Our project encountered several challenges and difficulties that required us to adapt, innovate, and overcome various obstacles. Below are some of the significant challenges we faced:

- **Limited Research Papers in Arabic ABSA:** A significant challenge we encountered in our project was the limited availability of research papers dedicated to Arabic Aspect-Based Sentiment Analysis (ABSA). This scarcity restricted access to established methodologies and insights, requiring us to adopt a multi-disciplinary approach and adapt methodologies from other languages. Overcoming this challenge allowed us to contribute to the field and provide valuable insights for future ABSA studies.

- **Overcoming Twitter's API Policies Change:** We faced a significant

challenge in our project when Twitter modified their API rules, making it hard to integrate the API into our system. This forced us to find a new solution and replace a substantial part of our project. Implementing the new solution was a major challenge as it required updating all project components to fit the new approach. We had to quickly adapt and modify our project to ensure a seamless integration of the new solution and minimize disruptions caused by the change.

- **Collecting Saudi Entertainment and Events Dataset:** The unavailability of a Saudi entertainment and events dataset presented a significant challenge for our project. It took much work to collect the data ourselves due to limited resources and time. However, we invested significant time and effort into data collection and faced various obstacles that hindered our progress. In the end, we managed to gather some data.

- **Challenge of Adapting Scope - Shifting Towards Hospitality:** The entertainment and events dataset annotation challenges mentioned in Chapter 5, section 5.3 forced us to reevaluate and adjust our scope. Shifting the focus to the hospitality industry required reviewing and updating our work. We aimed to ensure accurate and relevant analysis considering the new data landscape.

- **AI Models Integration:** One of the key challenges we confronted was integrating our three models for ABSA. In ABSA, a comprehensive integration of the three main tasks is significantly understudied and not thoroughly covered in existing literature. This gap presented a considerable obstacle in implementing such an integration accurately, particularly for a complex language like Arabic. The process proved to be both time-consuming and computationally demanding, primarily due to the inherent complexities involved in natural language processing. The time required for performing ABSA varied based on the data size in the file, thereby also impacting the users' experience.

8.2 Learned Skills and Lessons

Implementing multifaceted software projects that integrate various computer science areas is a challenging task that involves numerous skills. This project has enhanced our understanding and application of software engineering and machine learning in a unified setting. The skills utilized in this project encompass three main areas: technical, project management, and problem-solving.

1. **Technical Skills:** The project's technical requirements demanded the development and application of a range of specific skills:

    - Knowledge of various programming languages was essential for interpreting and producing code.
    - Acquired familiarity with several software development environments.
    - Gained expertise in defining functional and non-functional requirements.
    - Developed skills in designing system diagrams.
    - Demonstrated the ability to design and implement a system prototype.

2. **Project Management Skills:**

    - Teamwork: This project necessitated exceptional teamwork skills. Our team has developed effective communication, conflict resolution, reliability, respect for team dynamics, and a tolerance for trial and error learning.
    - Effective Communication: As technology enhances team connectivity, our team has developed the communication skills necessary for interaction in physical and virtual environments.
    - Planning: Our team has learned that careful planning is necessary for goal-setting, preparation, and achievement. Essential planning skills include deadline adherence, decision-making, goal-setting, scheduling, and timetable creation.
    - Effective Writing and Presentation: Our team refined writing and presentation skills as the project unfolded. We learned to concisely

and coherently convey ideas through organizing our ideas, drafting, revising, and editing written work. Additionally, we sharpened our presentation skills, making it possible to communicate these ideas to a diverse audience.

3. **Problem-Solving Skills:** Problem-solving is a crucial skill in the computer science field. Throughout the project, we encountered numerous issues that necessitated significant time investment in seeking and implementing solutions and exploring innovative approaches. This not only refined our problem-solving abilities but also improved our skills in debugging and error tracing. These experiences have enhanced our resilience and adaptability, which extend beyond computer science.

8.3 Future Work

1. **Models Improvements:**
   - Enhancing the Aspect Term Extraction (ATE) model, especially for multi-word aspect terms.
   - Training our ASC model on neutral sentiment class or other sentiment classes to expand the analysis of opinions.
   - Experimenting with multi-task ABSA by combining the ATE and ASC models into one.
   - Implementing a post-processing algorithm on the stemmed aspect terms for a better presentation of terms on the website.
2. **Performance Optimization:**
   - Improving the processing time of analyzing an uploaded file.
3. **Expansion and Integration:**
   - Branching out the analysis into the religious tourism and the entertainment and events tourism domains.
   - Integrating with Twitter's API or an available social media API for real-time analyses.

References

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. In *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 11–16. Association for Computational Linguistics.

Abdelgwad, M. M., Soliman, T. H. A., and Taloba, A. I. (2022a). Arabic aspect sentiment polarity classification using bert. *Journal of Big Data*, 9(1):1–15.

Abdelgwad, M. M., Soliman, T. H. A., Taloba, A. I., and Farghaly, M. F. (2022b). Arabic aspect based sentiment analysis using bidirectional gru based models. *Journal of King Saud University-Computer and Information Sciences*, 34(9):6652–6662.

Abdul-Mageed, M., Elmadany, A., and Nagoudi, E. M. B. (2021). ARBERT & 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)*, pages 7088–7105, Online. Association for Computational Linguistics.

Al-Ayyoub, M., Gigieh, A., Al-Qwaqenah, A., Al-Kabi, M. N., Talafhah, B., and Alsmadi, I. (2017). Aspect-based sentiment analysis of arabic laptop. In *ACIT'2017, The International Arab Conference on Information Technology*.

Al-Dabet, S., Tedmori, S., and Mohammad, A.-S. (2021). Enhancing arabic aspect-based sentiment analysis using deep learning models. *Computer Speech & Language*, 69:101224.

Al-Smadi, M., Qawasmeh, O., Talafha, B., and Quwaider, M. (2015). Human annotated arabic dataset of book reviews for aspect based sentiment analysis. In *2015 3rd International conference on future internet of things and cloud*, pages 726–730. IEEE.

Alasmari, W. A. and Abdelhafez, H. A. (2022). Twitter sentiment analysis for reviewing tourist destinations in saudi arabia using apache spark and machine learning algorithms. 18(3):215–226.

Alawami, A. (2016). Aspect terms extraction of arabic dialects for opinion mining using conditional random fields. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 211–220. Springer.

Alayba, A. M., Palade, V., England, M., and Iqbal, R. (2017). Arabic language sentiment analysis on health services. In *2017 1st international workshop on arabic script analysis and recognition (asar)*, pages 114–118. IEEE.

Alhajji, M., Al Khalifah, A., Aljubran, M., and Alkhalifah, M. (2020). Sentiment analysis of tweets in saudi arabia regarding governmental preventive measures to contain covid-19.

Aljabri, M., Chrouf, S. M. B., Alzahrani, N. A., Alghamdi, L., Alfehaid, R., Alqarawi, R., Alhuthayfi, J., and Alduhailan, N. (2021). Sentiment analysis of arabic tweets regarding distance learning in saudi arabia during the covid-19 pandemic. *Sensors*, 21(16):5431.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

AlSalman, H. (2020). An improved approach for sentiment analysis of arabic tweets in twitter social media. In *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–4. IEEE.

Alshammari, N. F. and AlMansour, A. A. (2020). Aspect-based sentiment analysis for arabic content in social media. In *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–6. IEEE.

Altowayan, A. A. and Elnagar, A. (2017). Improving arabic sentiment analysis with sentiment-specific embeddings. In *2017 IEEE international conference on big data (big data)*, pages 4314–4320. IEEE.

Ashi, M. M., Siddiqui, M. A., and Nadeem, F. (2018). Pre-trained word embeddings for arabic aspect-based sentiment analysis of airline tweets. In *International Conference on Advanced Intelligent Systems and Informatics*, pages 241–251. Springer.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bensoltane, R. and Zaki, T. (2021). Comparing word embedding models for arabic aspect category detection using a deep learning-based approach. In *E3S Web of Conferences*, volume 297, page 01072. EDP Sciences.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Boudad, N., Faizi, R., Thami, R. O. H., and Chiheb, R. (2018). Sentiment analysis in arabic: A review of the literature. *Ain Shams Engineering Journal*, 9(4):2479–2490.

Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649.

Dang, N. C., Moreno-García, M. N., and De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3):483.

Dautel, A. J., Härdle, W. K., Lessmann, S., and Seow, H.-V. (2020). Forex exchange rate forecasting using deep recurrent neural networks. *Digital Finance*, 2:69–96.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elmasri, R. (2021). Fundamentals of database systems seventh edition.

Fadel, A. S., Saleh, M. E., and Abulnaja, O. A. (2022). Arabic aspect extraction based on stacked contextualized embedding with deep learning. *IEEE Access*, 10:30526–30535.

Figma, D. T. (2022). Figma [version, 2022]. Avaliable at=`https://www.figma.com/`.

Gamal, D., Alfonse, M., El-Horbaty, E.-S. M., and Salem, A.-B. M. (2019). Twitter benchmark dataset for arabic sentiment analysis. *Int J Mod Educ Comput Sci*, 11(1):33.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Gupta, A., Sahu, H., Nanecha, N., Kumar, P., Roy, P. P., and Chang, V. (2019). Enhancing text using emotion detected from eeg signals. *Journal of Grid Computing*, 17(2):325–340.

Gupta, G. and Gupta, P. (2019). Twitter mining for sentiment analysis in tourism industry. In *2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)*, pages 302–306. IEEE.

Gupta, V., Lehal, G. S., et al. (2009). A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, 1(1):60–76.

Habash, N. (2022). Arabic natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 9–10.

Hu, X., Chu, L., Pei, J., Liu, W., and Bian, J. (2021). Model complexity of deep learning: A survey. *Knowledge and Information Systems*, 63:2585–2619.

Inzalkar, S. and Sharma, J. (2015). A survey on text mining-techniques and application. *International Journal of Research In Science & Engineering*, 24:1–14.

Jupyter, D. T. (2022). Jupyter notebook [version, 2022]. Avaliable at=`https://jupyter.org/`.

Kaur, C. and Kumar, V. (2015). Comparative analysis of iterative waterfall model and scrum. *International Journal of Computer Science Research (IJCSR) sve*, 3:11–14.

Kayid, A., Khaled, Y., and Elmahdy, M. (2018). Performance of cpus/gpus for deep learning workloads. *The German University in Cairo*.

Khurana, D., Koli, A., Khatter, K., and Singh, S. (2022). Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, pages 1–32.

Kung, D. (2013). Object-oriented software engineering. In *An Agile Unified Methodology*. McGraw-Hill Higher Education.

Liu, B. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge university press.

Microsoft, C. (2021). Microsoft visual studio code. Avaliable at=`https://visualstudio.microsoft.com/`.

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

Mohammad, A.-S., Al-Ayyoub, M., Al-Sarhan, H., and Jararweh, Y. (2015). Using aspect-based sentiment analysis to evaluate arabic news affect on readers. In *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, pages 436–441. IEEE.

Mohammad, A.-S., Qwasmeh, O., Talafha, B., Al-Ayyoub, M., Jararweh, Y., and Benkhelifa, E. (2016). An enhanced framework for aspect-based sentiment analysis of hotels' reviews: Arabic reviews case study. In *2016 11th International conference for internet technology and secured transactions (ICITST)*, pages 98–103. IEEE.

Muaad, A. Y., Davanagere, H. J., Guru, D., Benifa, J. B., Chola, C., AlSalman, H., Gumaei, A. H., and Al-antari, M. A. (2022). Arabic document classification: performance investigation of preprocessing and representation techniques. *Mathematical Problems in Engineering*, 2022:1–16.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., et al. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)*, pages 19–30. Association for Computational Linguistics.

Pozzi, F., Fersini, E., Messina, E., and Liu, B. (2016). *Sentiment analysis in social networks*. Morgan Kaufmann.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rana, R. (2016). Gated recurrent unit (gru) for emotion classification from noisy speech. *arXiv preprint arXiv:1612.07778*.

Saireddygari, S. R. (2021). Why nlp is the key to interpreting unstructured data in pharma drug discovery.

Slovikovskaya, V. (2019). Transfer learning from transformers to fake news challenge stance detection (fnc-1) task. *arXiv preprint arXiv:1910.14353*.

Talib, R., Hanif, M. K., Ayesha, S., and Fatima, F. (2016). Text mining: techniques, applications and issues. *International Journal of Advanced Computer Science and Applications*, 7(11).

Theobald, O. (2021). *Machine learning for absolute beginners: A Plain English introduction*. Amazon Digital Services LLC.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Virmani, C., Pillai, A., and Juneja, D. (2017). Extracting information from social network using nlp. *International Journal of Computational Intelligence Research*, 13(4):621–630.

Vision2030 (n.d.). Vision 2030. Avaliable at=`https://www.vision2030.gov.sa/ar/`.

Yang, X., Song, Z., King, I., and Xu, Z. (2022). A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zerrouki, T. (2010). pyarabic, an arabic language library for python. Avaliable at= `https://pypi.python.org/pypi/pyarabic`.

APPENDIX I

Questionnaire

- Questionnaire for tourists/visitors

  Question 1: What is your current role in the tourism industry? (ما هو دورك الحالي في صناعة السياحة؟)

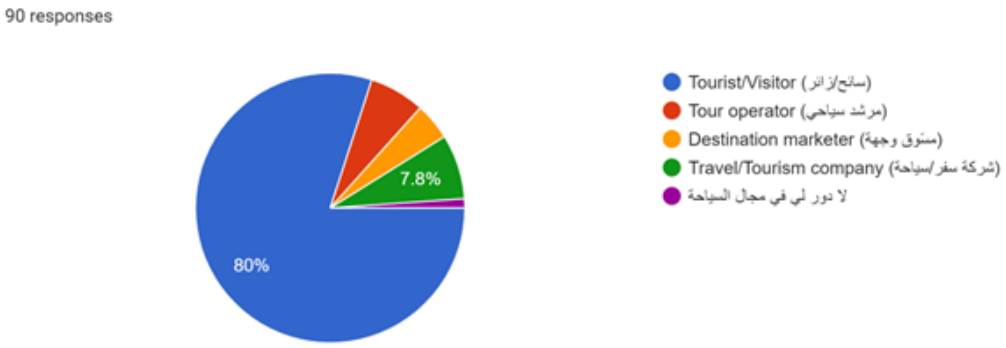

Figure I.1: Visitor: Question 1

Question 2: Do you utilize Twitter for researching and gathering information about tourism and entertainment destinations or events? (هل تستخدم تويتر للبحث وجمع المعلومات حول الوجهات أو الأحداث السياحية والترفيهية؟)

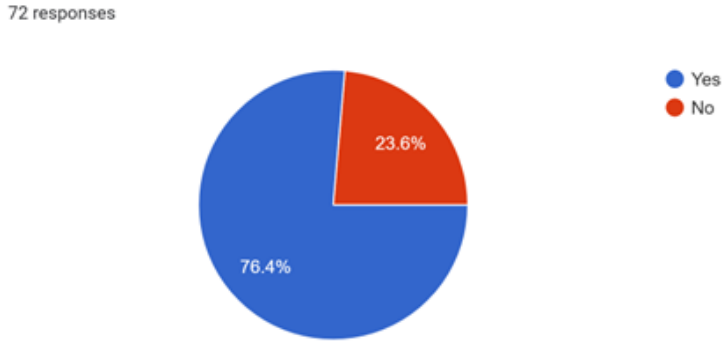

Figure I.2: Visitor: Question 2

Question 3: On a scale of 1-3, how important is it for you to know the public opinion on destinations or events before visiting them? (على مقياس من ١-٣ ، ما مدى أهتمامك بمعرفة الرأي العام حول الوجهات أو الأحداث قبل زيارتها؟)

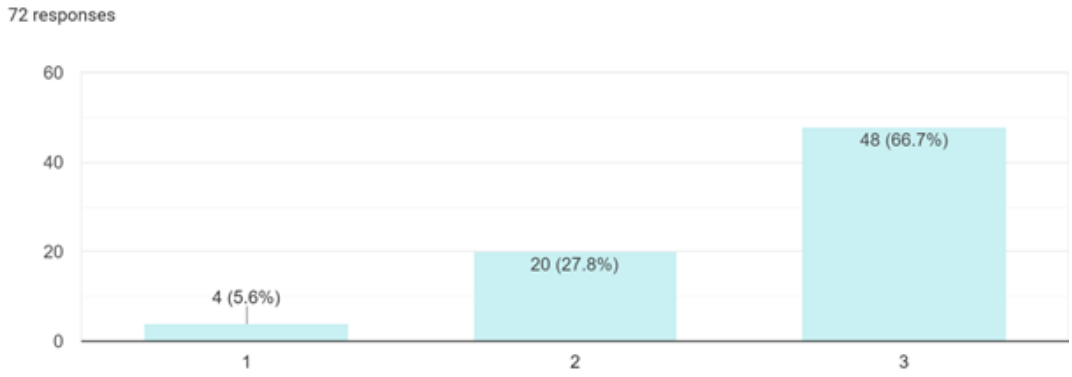

Figure I.3: Visitor: Question 3

Question 4: On a scale of 1-3, how difficult do you find it to obtain accurate feedback and opinions of others about Saudi tourism destinations and events? (على مقياس من ١-٣ ، ما مدى صعوبة الحصول على ملاحظات

دقيقة حول الوجهات والفعاليات السياحية السعودية؟)

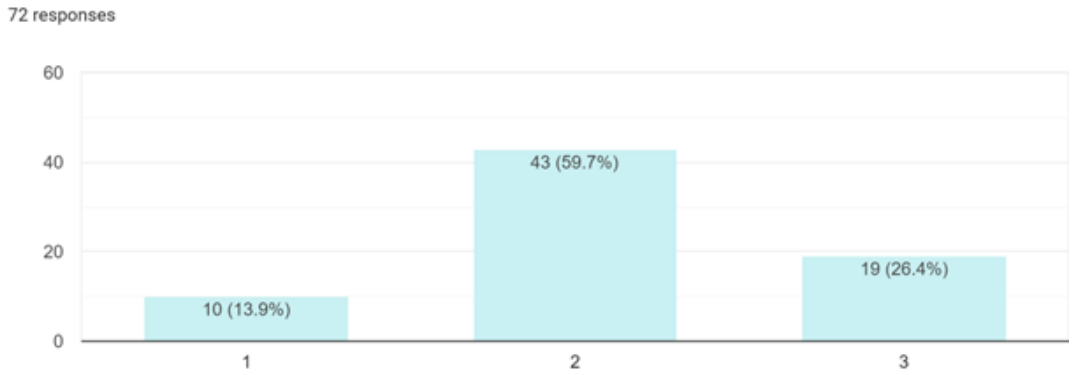

Figure I.4: Visitor: Question 4

Question 5: On a scale of 1-3, how satisfied are you with traditional methods (e.g., reading tweets) to find out what people think of the tourism industry?
على مقياس من ١-٣ ، ما مدى رضاك عن الأساليب التقليدية (مثل قراءة التغريدات) لمعرفة رأي الناس في صناعة السياحة؟)

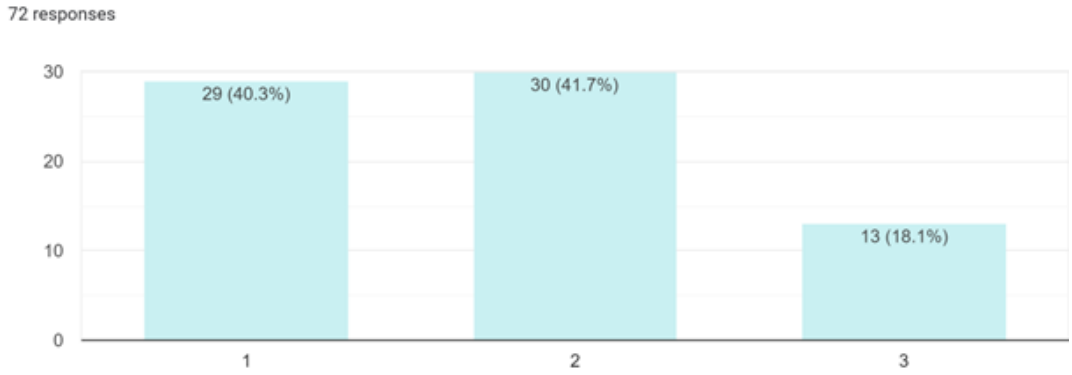

Figure I.5: Visitor: Question 5

Question 6: Would you like to use a web-based tool for analyzing public

opinions based on specific aspects of Saudi tourism on Twitter? (هل ترغب في استخدام أداة على شبكة الإنترنت لتحليل الآراء العامة في تويتر بناءً على جوانب معينة من السياحة السعودية؟)

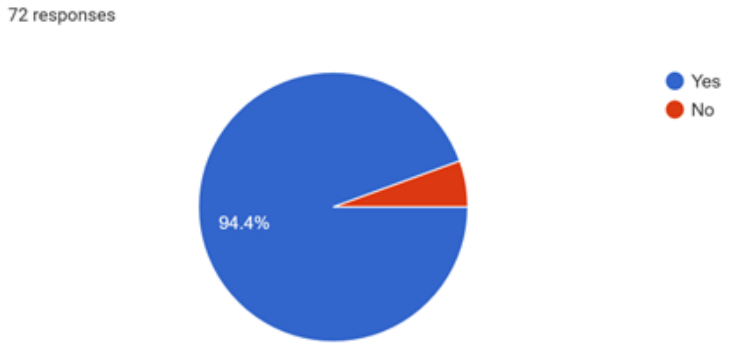

Figure I.6: Visitor: Question 6

Question 7: What aspects of a tourist or entertainment destination or event are most important to you when considering visiting? (Select all that apply) (ما الجوانب الأكثر أهمية لوجهة أو حدث سياحي أو ترفيهي بالنسبة لك عند التفكير في زيارته ؟ إختر كل ما ينطبق)

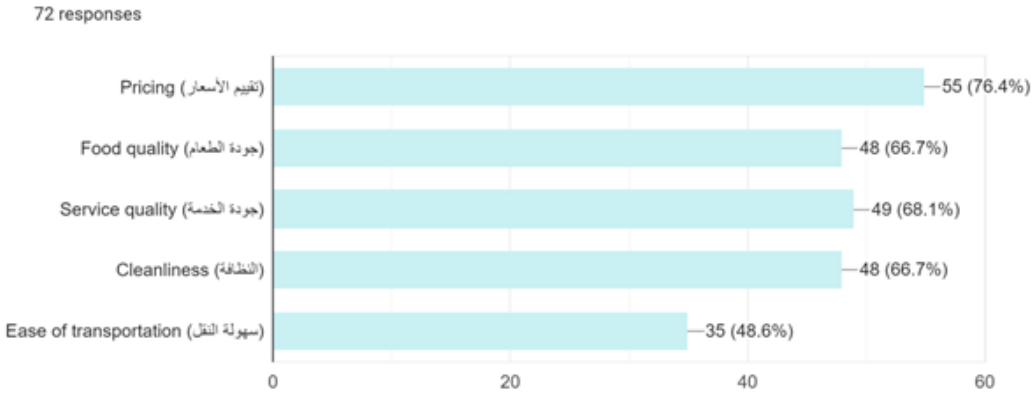

Figure I.7: Visitor: Question 7

Question 8: What features would you like to see in this tool? (ما هي الميزات التي تود أن تراها في هذه الأداة؟)

Figure I.8: Visitor: Question 8

- Questionnaire of Organizations

Question 1: What is your current role in the tourism industry? (ما هو دورك الحالي في صناعة السياحة؟)

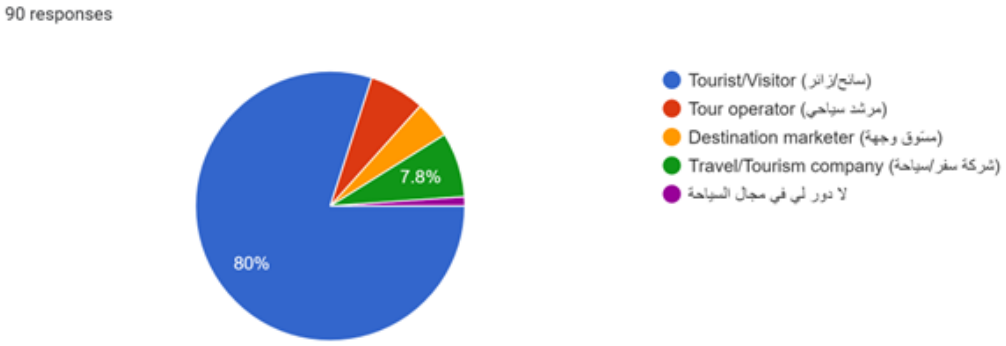

Figure I.9: Organization: Question 1

Question 2: Do you utilize Twitter for researching and gathering information about tourism and entertainment destinations or events? (هل تستخدم تويتر للبحث وجمع المعلومات حول الوجهات أو الأحداث السياحية والترفيهية؟)

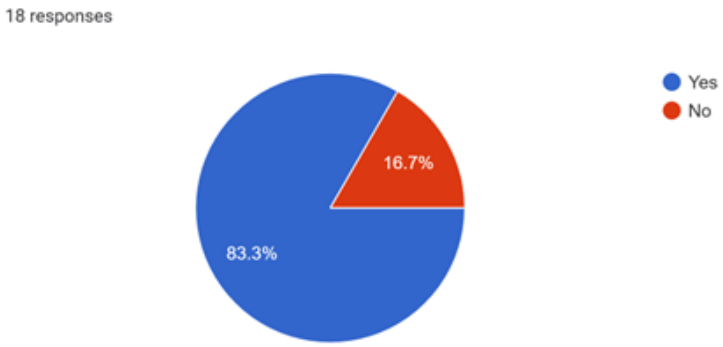

Figure I.10: Organization: Question 2

Question 3: On a scale of 1-3, how important is it for your organization to gather and consider customer feedback when making decisions about your

tourism offerings? (على مقياس من ١-٣ ، ما مدى أهمية قيام مؤسستك بجمع آراء العملاء ومراعاتها عند اتخاذ القرارات بشأن عروض السياحة الخاصة بك؟)

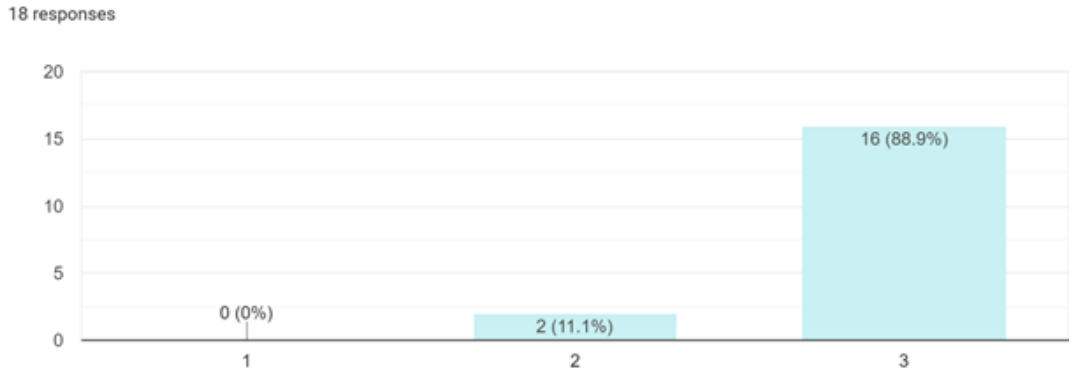

Figure I.11: Organization: Question 3

Question 4: On a scale of 1-3, how difficult do you find it to obtain accurate feedback about Saudi tourism destinations and events? (على مقياس من ١-٣ ، ما مدى صعوبة الحصول على ملاحظات دقيقة حول الوجهات والفعاليات السياحية السعودية؟)

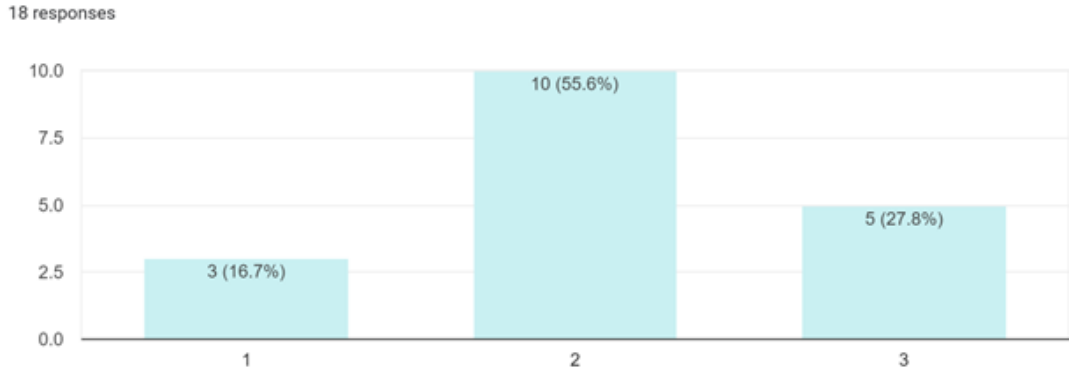

Figure I.12: Organizations: Question 4

Question 5: On a scale of 1-3, how satisfied are you with the traditional methods (e.g., conducting surveys or reading through tweets and identifying recurring themes or opinions) for gathering customer feedback in the

tourism industry? (على مقياس من ١-٣ ، ما مدى رضاك عن الأساليب التقليدية (مثل إجراء الدراسات الاستقصائية أو القراءة من خلال التغريدات وتحديد السمات أو الآراء المتكررة) لجمع ملاحظات العملاء في صناعة السياحة؟)

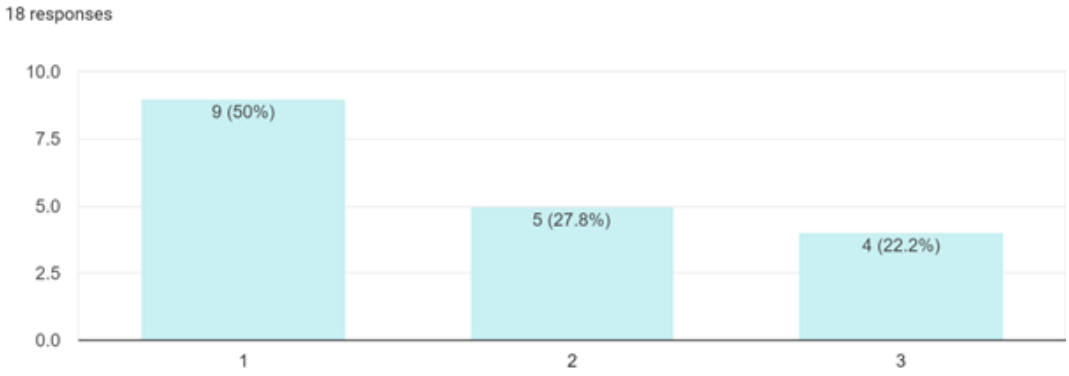

Figure I.13: Organizations: Question 5

Question 6: Would you like to use a web-based tool for analyzing public opinions based on specific aspects of Saudi tourism on Twitter? (هل ترغب في استخدام أداة على شبكة الإنترنت لتحليل الآراء العامة في تويتر بناءً على جوانب معينة من السياحة السعودية؟)

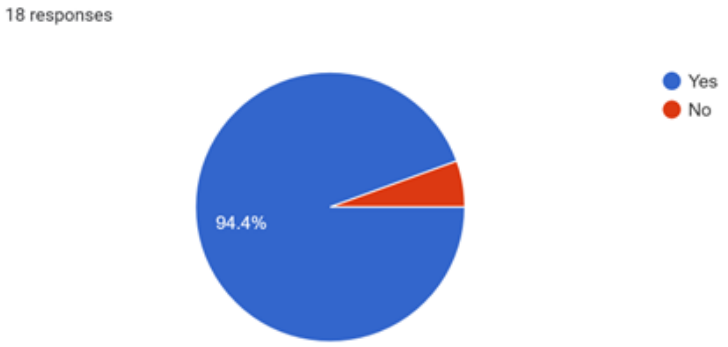

Figure I.14: Organizations: Question 6

Question 7: What aspects of a tourist or entertainment destination or event

are most important to you when considering promoting it? (ما الجوانب الأكثر أهمية لوجهة أو حدث سياحي أو ترفيهي بالنسبة لك عند التفكير في الترويج له؟ (اختر كل ما ينطبق)

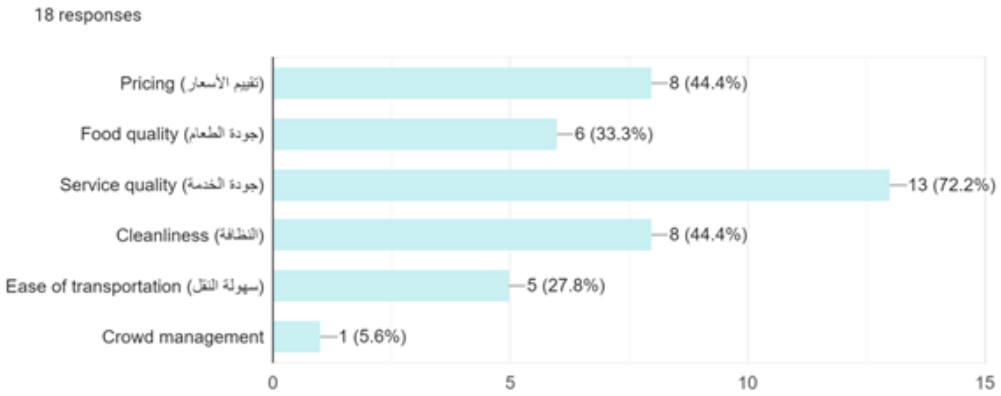

Figure I.15: Organizations: Question 7

Question 8: What features would you like to see in this tool? (ما هي الميزات التي تود أن تراها في هذه الأداة؟)

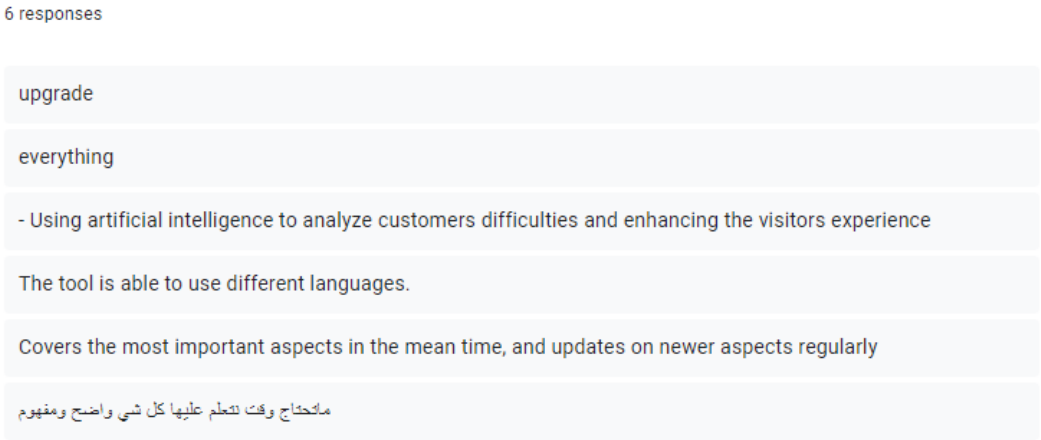

Figure I.16: Organizations: Question 8

APPENDIX II

Annotation Guidelines

1. **Aspect Term Annotation**:
    - Carefully read each tweet and identify explicit single or multiple terms that refer to specific aspects. Depending on the context, these terms can be either named entities, common nouns, or multi-word expressions.
    - Annotate only the first occurrence of a term in a sentence.
    - If the aspect term is ambiguous, discuss with other annotators.
2. **Sentiment Annotation**:
    - For each annotated aspect term, analyze the sentiment expressed in the tweet towards that term.
    - Assign a sentiment polarity (positive or negative) based on the context and tone of the tweet.
    - If the sentiment is unclear, consult with other annotators or the project team for guidance.
3. **Aspect Category Annotation**:
    - Carefully read each sentence in the customer review and recognize the entity (E) that the opinion is directed towards. Ensure that the identified entities are chosen from the predefined inventories provided for the domain.
    - Based on the context of the sentence, assign one or more appropriate attribute labels to the identified entities.

APPENDIX III

Front-end of Nire

Nire web-based tool main implemented interfaces:

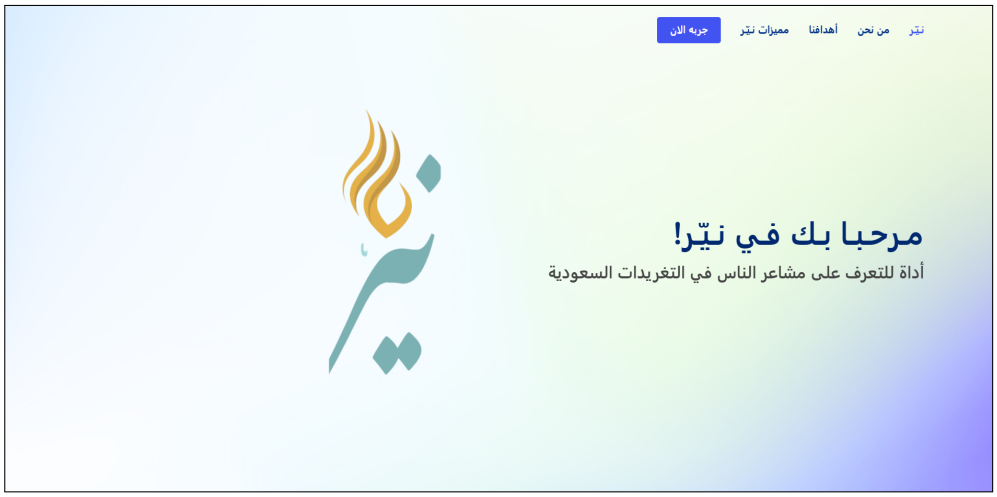

Figure III.1: Nire Landing Page.

Figure III.2: Nire Home Page.

Figure III.3: Nire Upload File Page.

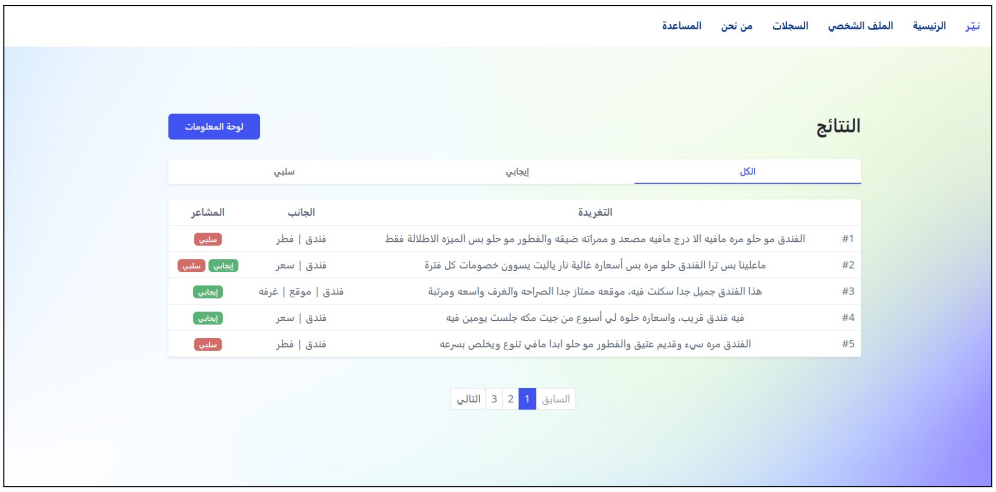

Figure III.4: Nire Table of the Result Page.

Figure III.5: Nire Dashboard Page.

Figure III.6: Nire History Records Page.

APPENDIX IV

Usability Testing Results

Usability Testing Survey

1. How easy was it to navigate the website and find what you needed?
    - Very easy
    - Moderately easy
    - Difficult
2. How would you describe your overall experience with Nire?
    - Satisfied
    - Moderately satisfied
    - Unsatisfied
3. Do you have any feedback?

Table IV.1: Participant 1 Usability Testing Results.

Task	Clicks	Duration	Task Completion	Autonomy Level
Task 1	2	54s	Completed	Independent
Task 2	2	18s	Completed	Independent
Task 3	7	21s	Completed	Semi-dependent
Task 4	15	1m, 16s	Completed	Independent
Task 5	16	25s	Completed	Independent
Task 6	1	10s	Completed	Independent
Task 7	2	16s	Completed	Independent
Task 8	3	30s	Completed	Independent
Task 9	3	10s	Completed	Independent
Task 10	1	10s	Completed	Independent

Table IV.2: Participant 2 Usability Testing Results.

Task	Clicks	Duration	Task Completion	Autonomy Level
Task 1	2	50s	Completed	Independent
Task 2	3	16s	Completed	Independent
Task 3	5	20s	Completed	Semi-dependent
Task 4	12	56s	Completed	Independent
Task 5	13	22s	Completed	Independent
Task 6	1	7s	Completed	Independent
Task 7	3	18s	Completed	Independent
Task 8	2	22s	Completed	Independent
Task 9	5	9s	Completed	Independent
Task 10	1	4s	Completed	Independent

Table IV.3: Participant 3 Usability Testing Results.

Task	Clicks	Duration	Task Completion	Autonomy Level
Task 1	2	1m	Completed	Independent
Task 2	2	39s	Completed	Independent
Task 3	5	30s	Completed	Semi-dependent
Task 4	10	1m, 31s	Completed	Independent
Task 5	11	19s	Completed	Independent
Task 6	2	10s	Completed	Independent
Task 7	3	22s	Completed	Independent
Task 8	2	26s	Completed	Independent
Task 9	3	15s	Completed	Independent
Task 10	1	4s	Completed	Independent

Table IV.4: Participant 4 Usability Testing Results.

Task	Clicks	Duration	Task Completion	Autonomy Level
Task 1	3	2m	Completed	Independent
Task 2	3	20s	Completed	Independent
Task 3	6	37s	Completed	Semi-dependent
Task 4	12	1m, 37s	Completed	Independent
Task 5	13	10s	Completed	Independent
Task 6	2	18s	Completed	Independent
Task 7	5	21s	Completed	Independent
Task 8	2	19s	Completed	Independent
Task 9	4	10s	Completed	Independent
Task 10	1	10s	Completed	Independent

Table IV.5: Participant 5 Usability Testing Results.

Task	Clicks	Duration	Task Completion	Autonomy Level
Task 1	2	1m	Completed	Independent
Task 2	3	23s	Completed	Independent
Task 3	6	36s	Completed	Semi-dependent
Task 4	12	1m, 40s	Completed	Independent
Task 5	13	5s	Completed	Independent
Task 6	1	31s	Completed	Independent
Task 7	3	51s	Completed	Independent
Task 8	3	33s	Completed	Independent
Task 9	3	8s	Completed	Independent
Task 10	1	3s	Completed	Independent

Table IV.6: Participant 6 Usability Testing Results.

Task	Clicks	Duration	Task Completion	Autonomy Level
Task 1	2	1m	Completed	Independent
Task 2	3	1m, 30s	Completed	Independent
Task 3	5	24s	Completed	Semi-dependent
Task 4	10	1m, 51s	Completed	Independent
Task 5	11	15s	Completed	Independent
Task 6	1	10s	Completed	Independent
Task 7	6	39s	Completed	Independent
Task 8	2	14s	Completed	Independent
Task 9	5	35s	Completed	Independent
Task 10	1	10s	Completed	Independent

Table IV.7: Participant 7 Usability Testing Results.

Task	Clicks	Duration	Task Completion	Autonomy Level
Task 1	4	1m, 29s	Completed	Independent
Task 2	3	19s	Completed	Independent
Task 3	5	25s	Completed	Semi-dependent
Task 4	10	3m	Completed	Independent
Task 5	11	22s	Completed	Independent
Task 6	1	10s	Completed	Independent
Task 7	2	16s	Completed	Independent
Task 8	3	18s	Completed	Independent
Task 9	4	10s	Completed	Independent
Task 10	1	5s	Completed	Independent