Steering a Standard Arab Language Processing Model Towards Accurate Saudi Dialect Sentiment Analysis Using Generative AI

Sulaiman Aftan* and Yu Zhuang*
Department of Computer Science
Texas Tech University
Lubbock, TX, USA
[0000-0003-2093-9894]

Ahmad O. Aseeri
Department of Computer Science, College of Computer Engineering & Sciences
Prince Sattam Bin Abdulaziz University
Al-Kharj, Saudi Arabia
[0000-0001-9863-4551]

Habib Shah
Department & College of Computer Science
King Khalid University
Abha, Saudi Arabia
[0000-0003-2078-6285]

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825944

***Abstract*—Sentiment analysis (SA) is crucial for many NLP applications across various domains. While Arabic is one of the world's major languages, high-quality NLP models developed for standard Arabic often underperform on regional dialects like the Saudi Dialect (SD) due to a lack of SD-specific training data. This paper presents a novel approach to adapting a high-resource language model, AraBERT, for low-resource dialect sentiment analysis by combining minimal SD data collection with generative AI. In the absence of openly accessible SD datasets, we augmented a small amount of collected SD data with GPT-generated SD data to fine-tune AraBERT for sentiment analysis in SD. Our contributions include (1) demonstrating the feasibility of low-effort data collection of a low-resource dialect for adapting existing high-resource NLP models and (2) leveraging GPT-generated data to augment collected data to enhance a high-resource language model for sentiment classification in a low-resource dialect, achieving significant improvements over the pre-trained high-resource model. These two contributions imply a potentially replicable approach that can serve as a template for future research in other low-resource NLP tasks. This paper presents a promising solution for enhancing model performance in low-resource dialects and has implications for similar under-resourced languages.**

Keywords— Generative AI, Sentiment Analysis, Saudi Dialect, NLP.

I. INTRODUCTION

Currently, more than seven thousand languages exist in this world. English and Arabic are among the most spoken languages, with an estimated speaker count of 1.5 billion and 310 million native and non-native speakers. Further, each language has various properties, characteristics, and accents or dialects [1], [2] A dialect is a linguistic phenomenon (LP) unique to a particular area of the world that is developed and shared by all people living there. The LPs are traits that pertain to examining how letters are pronounced and articulated, where the organs of pronunciation are placed for certain sounds, how soft sounds are scaled, how they are inflected, and how impacted neighboring sounds interact [3], [4]. If these traits are common in a region, they are referred to as dialects, and every area has a unique dialect that sets it apart from other places.

An Arabic dialect refers to a regional or social variety of the Arabic language that differs in pronunciation, vocabulary, idioms, expressions, meaning, and grammar from the standard or dominant form of the Arabic language. Dialects can develop due to geographical isolation, historical factors, cultural influences, and interactions with other languages. Among the different languages, English is also one of the most significant languages spoken in the world [5], [6]. English and Arabic are rich historical languages with deep cultural and linguistic histories. They differ significantly in their structure, writing systems, dialects, and usage contexts. Understanding these differences can dramatically enhance one's appreciation of both languages and the cultures they represent [7]. These dialects play an essential role in individual conversations, in the community, in tribes, in nationals, in customers, in business, in antisocial and social networks, and so on [6], [8]. Business and social networks are the most common platforms for using these dialects. Also, through these platforms, billions of datasets are generated daily, which makes it challenging to differentiate dialects from typical grammar. The differentiation between standard and dialect sentences of the Arabic language is difficult, especially for non-native speakers. Among the other computational, linguistics, and mathematical methods, technologies in artificial intelligence (AI) in general and machine learning (ML) and deep learning (DL) in particular, can provide automatic detection of the primary phase for speech recognition, which is the classification of Arabic dialect recognition handling the characterizing of the speakers' accents in any spoken language [6], [9].

The Arabic dialect, practically the Saudi dialect, is one of the most spoken Arabic languages and has more than 18 million speakers around the globe, especially inside Saudi Arabia, including the Middle East. Understanding the Saudi dialect is challenging, especially in various events, activities, behaviors, and other domains. The ML methods have been successfully applied in classifying these dialects and have achieved multiple levels of accuracy. For example, using 32063 tweets of Arabic dialect through LSTM, Bi-LSTM, and SVM, it was found that LSTM and Bi-LSTM have outperformed SVM with 94%, 92%, and 86.4 % accuracies respectively [10]. However, these typical ML methods can't accurately classify the differentiation between dialect and non-dialect sentences due to the vast dataset, deep meaning, morphological complexities, and other linguistics and scientific issues.

Researchers have proposed various DL methods for NLP tasks in the Arabic language. For example, AraBERT and MARBERT models built on Bidirectional Encoder Representations for Transformers (BERT) are highly successful in the standard Arabic language [11], [12], [13]. However, when we applied the AraBERT to sentiment analysis in the Saudi dialect, the results were unsatisfactory. Thus, the effectiveness of various Arabic NLP models remains unclear in the case of dialects.

To achieve good results for sentiment analysis in the Saudi dialect, we plan to fine-tune the AraBERT model with SD datasets. However, collecting a large amount of SD data and cleaning them into quality datasets suitable for NLP processing could take a lot of time and effort. Thus, we

* *These authors contributed equally to this work.*

propose to combine moderate effort in SD data collection with generative AI and use the combined data to fine-tune the AraBERT for SD sentiment analysis.

The remainder of this paper is organized as follows. A summary of the literature review of the SD is presented in section II. A description of our approach is presented in section III, followed by the simulation results in section IV. Finally, a conclusion of our findings is presented in section V.

II. Literature Review

Saudi Dialect (SD) identification is gaining significant attention in DL and Natural Language Processing (NLP) due to the growing use of dialectal Saudi Arabic text for both formal and informal communication on the web, events, and businesses, requiring research on dialectal corpora and language classification and identification. The SD which are common in a few dialects are the Hijazi Dialect (famous and generated from the western part of KSA), Najdi Dialect (famous and generated from the middle part of KSA), Janobi Dialect, and Hasawi Dialect, also called the middle part of Saudi Arabia), Hijazi (the western part of Saudi Arabia), Gulf Arabic (the eastern part of Saudi Arabia), and southern dialects (the southern part of Saudi Arabia), which share the same Arabic characters as other dialects that are spoken in close geographical regions such as Yemen, Egypt, UAE, Qatar, and Bahrain, without following and grammatical rules, unlike MSA[14]. For this purpose, various researchers have applied cutting-edge technologies, including DL models, for multiple tasks such as generating, classifying, and recognizing SD. Various ML models are used to classify the dialect from the given large dataset and predict the dialect based on the trained dataset, with multiple accuracy ratios[10], [15].

Research has used DL techniques for Arabic dialect annotation and sentiment analysis. For the Multi-Dialect Arabic Sentiment Twitter Dataset, the SVM and LSTM were successfully used, and the Arabic dialect dataset was extracted from Twitter of two countries, UAE and Egypt [8], [9]. The maximum accuracies reached 70 and 64.8 in the Egyptian and UAE dialects, respectively, for automatic dialect classification of the famous dataset Arabic Online Commentary (AOC), labeled with Egyptian, Gulf, Iraqi, Levantine, Maghrebi, MSA, and some other categories for multiple dialects or others. The four ML classifiers are long-short-term memory (LSTM), convolutional neural networks (CNN), bidirectional LSTM (BLSTM), and convolutional LSTM (CLSTM) were simulated for classification purposes, where the maximum accuracies obtained by LSTM was 71.4% [9]. In 2020, Alahmary et al. utilized DL techniques to develop a sentiment analysis model for the SD, using Convolutional Neural Networks (CNNs) to classify sentiment from tweets, demonstrating superior performance compared to traditional ML like SVM and NB methods [16].

For sentiment analysis, Almuqren L used AraCust for 20,000 tweets. The study offers comprehensive information on the corpus's construction, preprocessing, annotation processes, and features. AraCust and AraBERT were found to be superior when compared with typical NLP models. They achieved the goal of predicting customer satisfaction of telecom companies based on Twitter analysis of the 41.63 million subscribers [17], [18]. The study presents a dataset for sentiment analysis of the Saudi dialect, highlighting its potential for future deep-learning applications. The dataset, curated from tweets, was found to be valuable in training models, demonstrating its importance [15].

However, these models are sometimes unreliable due to lacking linguistic domain experts, which may reduce their interpretability and effectiveness, especially in decision-making. Therefore, new AI models called Explainable AI are used to provide the output under the deep consideration of the domain linguistic expert, which visualizes the obtained results for the correct decision in various tasks such as classification or predictions [19]. Very little research has been conducted in the XAI and Saudi Dialect domains, mainly since no new SD dataset has been generated and simulated with these models.

The study [20] introduces an explainable Arabic sentiment classification framework by introducing a Gaussian noise layer in DL models like BiLSTM and CNN-BiLSTM. This reduces overfitting and improves performance. The paper also presents a locally explainable surrogate model, LIME, in Arabic Sentiment Analysis (ASA) for the first time, providing easy-to-understand explanations for sentiment predictions. The work [21] uses Local Interpretable Model-agnostic Explanation (LIME) to predict sentiment polarity in Arabic texts about LASIK surgery. The LSTM achieves an accuracy of 79.1% on the proposed dataset, demonstrating accurate results on specific words contributing to sentiment classification. The results are validated by comparing word count with probability weights in the context of ASA.

To the best of our knowledge, no work exists on the annotation and sentiment analysis of Saudi Dialect (SD). The lack of research articles in the SD causes a significant gap in data availability. Therefore, we will explore the power of generative AI using the GPT model to generate a new dataset to overcome this issue. This will improve DL models such as AraBERT in complex tasks like sentiment analysis.

III. Proposed Approach

A. The Architectural of the SD Sentiment Analysis Model

To develop an NLP model for an application for a low-resource language (LRL), our proposed approach consists of the following steps:

1. To leverage the power of transferring learning, we choose an existing well-trained NLP model for the same type of application but for a close and higher-resource language (HRL), which is to be fine-tuned with data in the low-resource language into a low-resource language model.
2. To reduce effort and time in collecting LRL data and cleaning them into processable format, we plan to combine data collection/cleaning with generative AI, where the cleaned collected data will be fed to the generative AI model to generate synthetic LRL data.
3. The cleaned collected data to be used for fine-tuning the HRL Model into an LRL Model needs to be LRL data. Since collected data may contain words and phrases in the low-resource language and words and phrases not in the low-resource language, an annotator or filter is needed that can filter out words/phrases not in the low-resource language, and an appropriate NLP model needs to be developed for this annotator/filter.
4. The training of the generative AI model that is needed in Step 2 for generating synthetic LRL data needs an annotator to label each generated data item to gradually train the generative AI model into one of high accuracy

in generating LRL data, and the annotator/filter developed in Step 3 meets the purpose.

5. The annotator/filter used in Step 4 to label each generated data item is also used to select LRL data, which are placed into the dataset consisting of cleaned collected data and generated data for fine-tuning the HRL model into the LRL model.

Our general approach for fine-tuning an HRL model into an LRL model is presented above. To develop our specific sentiment analysis model in the Saudi dialect, we need to find an appropriate model (described in Step 1) in an HRL close to the Saudi dialect. The AraBERT and MARBERT are two candidate models. The AraBERT was pre-trained in standard Arabic, and the MARBERT was pre-trained in standard Arabic and other Arabic dialects, including the Saudi dialect. However, the MARBERT is a small model, and we fine-tuned it for our sentiment analysis, but the results were far below expectations. We then tried the AraBERT, a large language model, and found it fine-tunable for sentiment analysis in the Saudi dialect. So, the AraBERT is chosen as the HRL model.

While the MARBERT was unsuitable for our sentiment analysis in the Saudi dialect, we found it functioning well as the annotator/filter (described in Steps 3-5) after being fine-tuned with a small amount of Saudi dialect data, and it was hence chosen. For the generative AI model (described as part of Step 2), we choose a GPT2 model, the AraGPT2, for generative synthetic Saudi dialect data.

For the data collection (described as part of Step 2), we used the X (formerly Twitter) data. Collected data was then preprocessed, consisting of three tasks: cleaning the data, normalization, and tokenization.

Our development of the Saudi dialect sentiment analysis model, following the general approach for fine-tuning an HRL model into an LRL model, is illustrated in Figure 1.

The following subsections discuss the collected dataset, data preprocessing, examples and descriptions, performance metrics, MARBERT, and AraBERT.

B. The Collected X Dataset

In the Arab world, platforms like X (formerly Twitter), Facebook, Instagram, and YouTube are widely used as prominent social networking platforms[22]. A study found that most researchers utilized X (formerly Twitter) and Facebook to gather Arabic text datasets[23]. According to a Saudi Communications, Space, and Technology Commission (CST) report, social media is essential in Saudi Arabia due to high internet usage and a young population. The number of users on X (formerly Twitter) is more than 25 million, and it is considered the third most visited social media platform in Saudi Arabia, as indicated in Figure 2 [24].

To gather tweets from Saudi Arabia using the X API v2, we implemented an organized methodology that included multiple essential stages. Initially, we established our environment by importing essential libraries, creating codes for authentication and URL generation, and establishing a connection to the X API endpoint. Our primary objective was to locate tweets that contained specified Arabic and dialect keywords while ensuring that only tweets originating from Saudi Arabia were included.

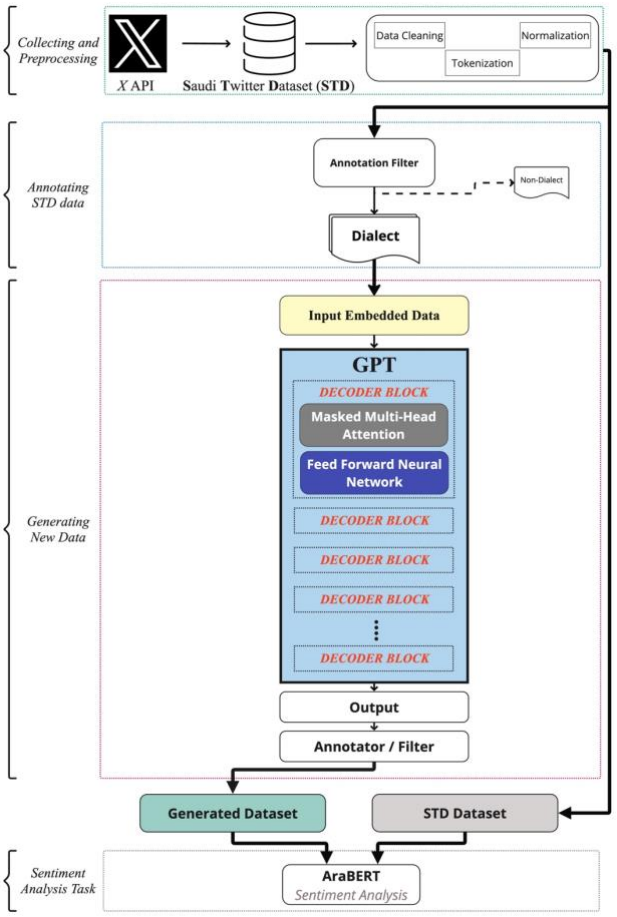

Figure 1 The Architecture of the SD Sentiment Analysis Model

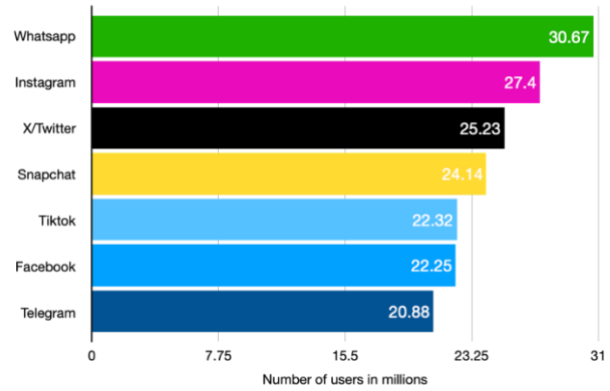

Figure 2 Most Visited Social Media Platforms in Saudi Arabia

We utilized the Google Colab platform coupled with NVIDIA's T4 GPU hardware. This hardware enhances the performance of several cloud applications, including high-performance computing. In addition, we employed Tweepy, a Python library that offers an interface for accessing X's API. Initially, we verified our access to the X API using a Bearer token, which we securely saved in environment variables. Next, we determined the keywords and phrases pertinent to our search and specified 'SA' in the country-code box to denote Saudi Arabia. In addition, we ensured that the language option was set to Arabic and implemented a filter to restrict the search to tweets geotagged from Saudi Arabia.

We established a function to generate the API URL by incorporating relevant query parameters. These parameters encompassed expansions to collect additional tweet data and fields that indicated the tweet content of our interest, such as text and language. Initially, we incorporated a geo-bounding box to restrict the results to Saudi Arabia.

In order to manage the rate constraints set by the X API, we devised a mechanism that executed a loop to send requests in groups, with intervals between each request to prevent surpassing the limit. We observed the existence of a 'next_token' in the API answers to navigate through the results, guaranteeing that we gathered a maximum number of relevant tweets. We analyzed each valid answer to retrieve tweet data, which we added to a CSV file for further analysis.

The tweets were collected between January 2024 and June 2024 and contained around 50,000 Arabic tweets. While the majority of users included useless geographical information, a significant portion did identify their specific city or region. A comprehensive investigation was conducted to determine the phrases related to Saudi Arabia, such as 'KSA' and 'Saudi,' the names of Saudi provinces and cities, and important hashtags.

During the process, we incorporated debugging print statements to validate the accuracy of our URL generation and to resolve any errors detected while making API requests. Using a systematic approach, we gathered a large dataset of tweets from Saudi Arabia that included our chosen keywords. This ensured we adhered to X's usage regulations and effectively managed any possible problems. Table 1 illustrates a description of our collected corpus of the Saudi dialect. We named our corpus STD, which stands for Saudi Twitter Dataset.

Table 1 Description of the Saudi Twitter Dataset (STD)

Number of Tweets	50,000
Number of Words	1,320,788
Language	Arabic
Platform	X

C. Annotating the X Dataset

1) Preprocessing

The preprocessing stage refers to the process of cleansing the data to minimize errors and enhance the performance of semantic analysis. Furthermore, text preprocessing is crucial in developing word embedding models since it substantially influences the ultimate outcomes[25]. Preprocessing encompasses various stages, including tokenization, removal of URLs, punctuation marks, digits, unnecessary whitespace, user mentions, hashtags, and emojis[26]. In this research paper, we implemented the following preprocessing steps using Python's NLTK (Natural Language Toolkit) library, addressing the Arabic script by eliminating diacritics, removing Tatweel symbols, and rectifying prevalent spelling errors. By doing this preprocessing step, we guaranteed that our dataset was devoid of mistakes or inconsistencies, making it appropriate for training the MARBERT model. Table 2 illustrates an example of the preprocessing steps applied to the STD corpus. By utilizing these strategies, we successfully obtained over 27,870 tweets. These tweets contained an overall sum of 379,974 words written in the Arabic language.

Table 2 Example of the Preprocessing Steps on the STD Corpus

Preprocessing Steps	Example Before Processing	Example After Processing	Explanations
Removing URL	@LWHLH_ لا لا لأ ياهؤه ، دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!! 🤣🤣🤣🤣🤣 HTTPS://TWITTER.COM/EKLMRE/STATUS/999684853468626944	@LWHLH_ لا لا لأ ياهؤه ، دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!! 🤣🤣🤣🤣🤣	We excluded external website links from tweets to maintain focus on textual content.
Removing English letters, user mentions, and hashtags	@LWHLH_ لا لا لأ ياهؤه ، دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!! 🤣🤣🤣🤣🤣	لا لا لأ ياهؤه ، دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!! 🤣🤣🤣🤣🤣	In focusing on the Saudi Arabic dialect, we filtered English words from the tweets to ensure the model concentrated solely on Arabic content.
Removing punctuation	لا لا لأ ياهؤه ، دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!! 🤣🤣🤣🤣🤣	لا لا لأ ياهؤه دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!! 🤣🤣🤣🤣🤣	We removed punctuation marks from the tweets to standardize the text and aid in dialect classification.
Removing emojis	لا لا لأ ياهؤه دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!! 🤣🤣🤣🤣🤣	لا لا لأ ياهؤه دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!!	Emojis have been removed from the text to prevent them from affecting the language model's understanding.
Removing whitespace	لا لا لأ ياهؤه دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!!	لا لا لأ ياهؤه دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!!	Whitespaces has been removed to improve model performance.
Removing Arabic diacritics	لا لا لأ ياهؤه دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!!	لا لا لا ياهوه دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!!	We removed diacritics from the Arabic script to standardize the text and improve model performance.
Removing Arabic Tatweel	لا لا لا ياهوه دايخه كنت بناااام بس من كثر الضحك خلاص صحصحنا!!!!!	لا لا لا ياهوه دايخه كنت بنام بس من كثر الضحك خلاص صحصحنا	We standardized the text by removing Tatweel characters, used in Arabic script for stylistic stretching, to avoid unnecessary variation.
Removing consecutive duplicate Arabic words	لا لا لا ياهوه دايخه كنت بنام بس من كثر الضحك خلاص صحصحنا	لا ياهوه دايخه كنت بنام بس من كثر الضحك خلاص صحصحنا	We removed consecutive duplicate Arabic words to improve data quality and language pattern learning by the model.

2) Applying the MARBERT model for Saudi Dialect Annotation

We compiled a comprehensive list of terms and phrases for the methodology's core, including the Saudi dialect and Modern Standard Arabic (MSA) non-dialect. This list was crucial for the task of Saudi dialect annotation. We leveraged a pre-trained MARBERT model for the accomplishment of this task since MARBERT is a large, masked language model (MLM) that focuses on both (MSA) and Arabic dialects [27]. The MARBERT model was built on the pre-trained BERT (Bidirectional Encoder Representations from Transformers) model[13]. BERT is designed to comprehend the contextual meaning of words within a sentence by considering the words that come before and after them. The model utilizes the Transformers library, which provides resources and pre-trained models for tasks related to natural language processing (NLP). These tweets were tokenized using a WordPiece tokenizer that adheres to the BERT model, providing that the text is divided into tokens to allow the model to understand and handle it with clarity. In addition, we utilized the PyTorch package to train the text classification model. We believe the MARBERT model can effectively process and comprehend the Saudi tweets.

During fine-tuning, the model learned to distinguish between dialectal and non-dialectal tweets by recognizing patterns in the supervised data. Hyperparameters such as the learning rate, batch size, and number of training epochs have been adjusted.

Then, the model is evaluated on the testing data, which were set to be 20% and 80% for training data. These evaluations provide insight into the model's generalization ability to unseen data. An evaluation metrics of the model performance, such as accuracy, precision, recall, and F1-score, were conducted. Also, an early stopping was implemented to prevent overfitting or underfitting of the data, and an error analysis was performed to detect the usual patterns of errors made by the model. This analysis helped us to identify the limitations or specific challenges of the model in the Saudi dialect annotation, such as code-switching between MSA and the Saudi dialect in a tweet. These evaluations on the unlabeled dataset encompassed the model to predict each tweet, whether in the Saudi dialect or not. These predictions have been labeled and saved as an annotated dataset, and they are ready to be used in the next step, which is applying the GPT model.

D. Applying the GPT model for Saudi Dialect Generation

To further enhance our dataset with dialectal text, a Generative Pre-trained Transformer (GPT) model was applied to the annotated data[28]. An AraGPT2 model was chosen for this task due to the model's capabilities in generating Arabic text[29]. The core of this AraGPT2 model is 12 layers (blocks) of a transformer decoder architecture responsible for processing the input data through these layers to create meaningful output. AraGPT2 contains 1.46 billion parameters, which allows the model to capture the complexity of the Saudi dialect variation and enhance its ability to generate coherent and accurate text from the annotated data of Saudi dialect tweets.

We started this implementation using a cloud platform, Google Colab, with NVIDIA T4 GPU. After installing the required libraries, we passed the annotated dataset to include only the tweets labeled as dialects by the MARBERT model. During this process, we used a Byte Pair Encoding (BPE) tokenizer to handle large vocabulary and rare words by converting text into tokens. These token IDs were converted into dense vectors with a size of 768 units by the embedding layer and used as input into the model. A positional encoding is added to these tokens to provide more information on each token position in the sequence. In order to prevent overfitting of the embedded data, a random dropout layer of 0.1 is applied to deactivate some of the neurons during the training. These tokens went through a list of 12 blocks, which are the core of the GPT model. Each block contains layers, as described below.

A normalization layer is applied before the attention mechanism, which stabilizes and accelerates the model training. An attention mechanisms layer allows the model to predict the next token in the order given by the previous token. The feed-forward networks play crucial roles in enhancing the model learning capabilities, which adds non-linearity and helps the model capture the complexity of the SD. These steps were repeated iteratively to generate subsequent tokens until the end of the sequence token. The output of these processes passed through a SoftMax layer that converted the raw data into probabilities over the vocabulary. To make sure all the production of this model is relevant to the given prompt, a layer of annotation has been added to the pipeline to filter out irrelevant or poorly generated text, as shown in Figure 3[30].

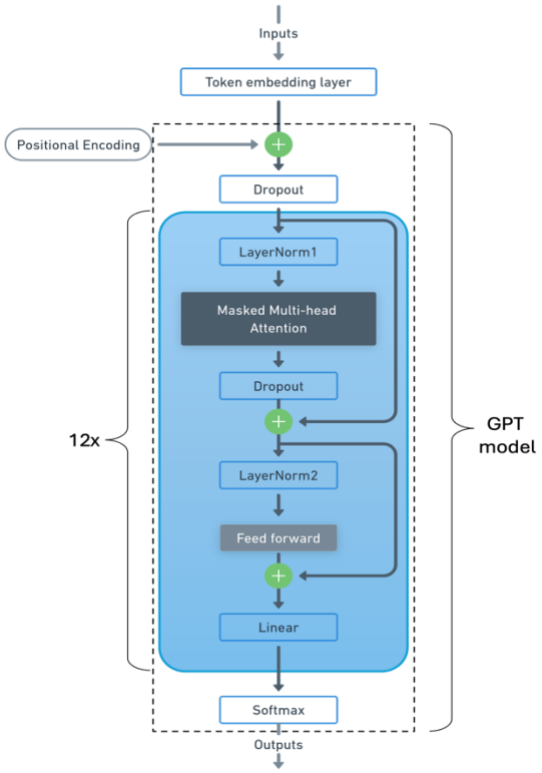

Figure 3 AraGPT Model Architecture

Also, to evaluate the model performance, we used two metrics: perplexity, which measures how well the model predicts the tweet and the BLEU (Bilingual Evaluation Understudy) score, which measures the similarity between the generated tweet and the original tweet[31], [32] By carefully optimizing the training techniques and hyperparameters such as the number of epochs, batch size, and learning rate, and monitoring the quality of the metrics, we were able to improve the model performance and ensure that the model generated tweets that were coherent and accurate.

E. GPT Generated Dataset

The fine-tuning of the AraGPT2 model on the STD showed surprising results where the model could generate human-like text while maintaining the coherent and syntactical sound of the Saudi dialect. This demonstrates that the model learned structures and words that are particular in the Saudi dialect from the training data. Table 3 shows a description of the generated Saudi dialect dataset.

Table 3 Description of the Generated Dataset

Number of Generated Tweets	19,251
Total Characters in Generated Data	474,2250
Average Characters per Generated Tweet	246.34
Total Words in Generated Data	95,0627
Average Words per Generated Tweet	49.38

Table 4 illustrates examples of the generated Saudi dialect dataset, where red is the original tweet "prompt," and green is the model-generated text. Following each example are BLEU scores and perplexity scores. The model demonstrated the capability to generate human-like text and produce lengthy, coherent, and task-specific content.

Table 4 Examples of the Generated Saudi Dialect Dataset

Example 1	***Original Tweet***	هالصوره قصده لا تسوي نفسك قوطي ما تفهم
	Generated Saudi dialect	هالصوره قصده لا تسوي نفسك قوطي ما تفهم ولا تشخصن على الناس اللي انت قاعد تسمع كلام في تويتر، ترا حتى لو كان عندك حساب او شخص ماخذها من يوتيوب مثلا. **BLEU Score:** 0.71 **Perplexity Score:** 17.34
Example 2	***Original Tweet***	شيء طبيعي لازم كان فيه بدائل استثمارية مثل ما قلت لك
	Generated Saudi dialect	شيء طبيعي لازم كان فيه بدائل استثمارية مثل ما قلت لك انت مثلا في شركات يعني لو عندك مشاريع زراعية او انتاج حيواني بس انا اتوقع هذا من القطاعات اللي قاعده فيها هذه الشركات حتى الان على سبيل المثال. **BLEU Score:** 0.85 **Perplexity Score:** 20.30
Example 3	***Original Tweet***	كلن له وجهة نظر
	Generated Saudi dialect	كلن له وجهة نظر، حتى لو كان ما يفكر في نفسه عشان يعرف هو اللي بيصيرله زي كذا، يعني ليه لما يشوفها بشكل مختلف عن غيره بيكون. **BLEU Score:** 0.88 **Perplexity Score:** 10.55
Example 4	***Original Tweet***	وش دخلك
	Generated Saudi dialect	وش دخلك في الموضوع؟ ترى هذا كلام من حقين الرجال وش تحس فيه؟ يعني انت الحين قاعد تقول لنا لا تشخصن النساء. **BLEU Score:** 0.92 **Perplexity Score:** 12.88
Example 5	***Original Tweet***	راعي السيارة غلطان ما فيها كلام
	Generated Saudi dialect	راعي السيارة غلطان ما فيها كلام الا صوته واضح انه راعي الدباب مو داق اشارة ولا غيره صاحب السياره غلط في كلامه خصوصا لو كان الطريق طبيعي ان هذا مخالفه من اليمين. **BLEU Score:** 0.93 **Perplexity Score:** 08.20

F. Applying the AraBERT model for Sentiment Analysis

The study of sentiment analysis has gained significant attention in Arabic NLP, with several datasets and methodologies. One of these successful methodologies is the AraBERT model. AraBERT is a pre-trained language model explicitly created for the Arabic language that uses the BERT (Bidirectional Encoder Representations from Transformers) architecture and configuration. This model has 12 encoder blocks, 768 hidden dimensions, 12 attention heads, and ~110 M parameters. Its pre-training on a large corpus of Arabic text, such as social media content, news stories, and Wikipedia entries, makes it ideal for comprehending and processing Arabic language nuancedly[12].

In this research, we conducted a comparative analysis of the AraBERTv02 model's performance in sentiment analysis across multiple datasets, including our dataset Saudi Twitter Dataset (STD), the AraCust dataset, the generated dataset of STD merged with the AraCust dataset, and we combined STD dataset and AraCust dataset. AraCust is a dataset for analyzing customer feedback on Saudi telecom companies. This dataset was collected from eight Twitter accounts from January until June 2017, and a number of 20,000 tweets were written in the Saudi dialect. These tweets were manually labeled as 32% positive and 68% negative[17].

We followed the same preprocessing steps that were applied to the STD dataset for all the datasets to make sure that the datasets were cleaned, normalized, and tokenized before passing them to the model. After the preprocessing was applied to all datasets, the number of data points was reduced. After that, datasets were fine-tuned using the same hyperparameters: a maximum sequence length of 128, a batch size of 16, the model is trained for four epochs, an AdamW optimizer with a learning rate of 2e-5, the mixed precision data type "FP16" for gradient computations, class weights, and early stopping ensures the model is well-suited to handle class imbalance and stops training optimally. All the experiments have been performed on Google Colab L4 GPU, and the datasets were divided into 80% for training and 20% for validation. In contrast, the same validation set was used to evaluate the model performance using precisions, recall, and F1-scores on positive and negative classes. Also, an accuracy metric on the whole dataset and a confusion matrix are used to visualize the model's performance on the classes.

IV. Simulation Results

Here are the simulation results obtained by MARBERT, AraGPT, XAI, and AraBERT of the datasets for annotation and sentiment analysis with various performance matrices.

A. MARBERT model performance

The MARBERT model was fine-tuned for five epochs using a batch size of 32 for training and evaluation. By fine-tuning the MARBERT model with labeled data, we empowered the model to differentiate between dialect and non-dialect phrases and words in the given list. The evaluation findings of the MARBERT model were exceptional, achieving an accuracy of 98%. Precision and recall showed how the model could correctly identify the characteristics of Saudi dialects. Table 5 shows the evaluation results of the model. This level of performance indicates that the model is well-suited for accurately identifying Saudi dialects.

Table 5 The evaluation metrics of the MARBERT model

Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-Score
0.0020	0.095	0.981	0.969	0.989	0.979

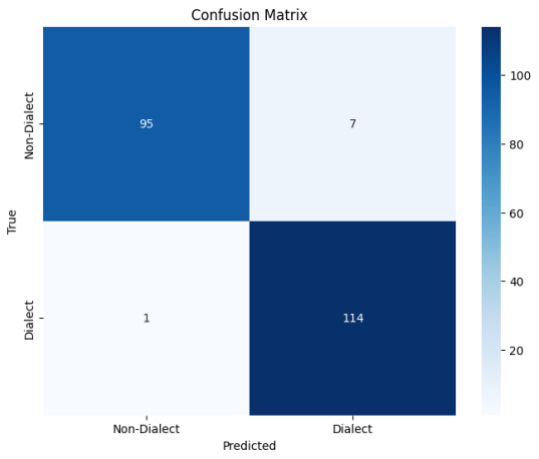

Figure 4 Confusion Matrix on the Annotated Dataset

Moreover, the results confirmed the MARBERT model's ability to accurately differentiate between dialect and non-dialect phrases and words, as shown in the confusion matrix in Figure 4.

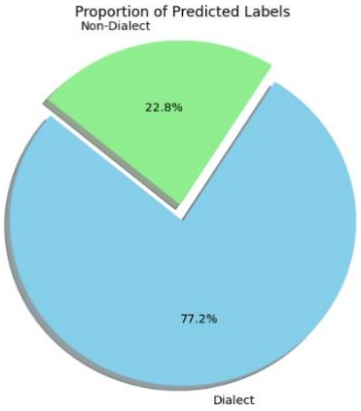

Figure 5 Predicted Label Distribution on the Dataset

Through fine-tuning, the MARBERT model understood the distinct features of the Saudi dialect, allowing it to make precise predictions and annotate tweets. To our knowledge, this is the first time a deep learning model (DL) has classified and annotated a dataset without human assistance. After completion, we found that 19,251 tweets were classified as dialect, 77.2% of the dataset, and 8,620 as non-dialect, 22.8% as shown in Figure 5.

1) eXplainable AI (XAI) on MARBERT

Understanding the complexity and output of a machine learning model is crucial. eXplainable AI (XAI) is a set of tools and techniques that explain the results and decisions of the machine learning model [33]. The main goal of XAI is to improve our understanding of the model's performance and to trust the result of this model. Since The MARBERT model has been successfully annotated, the Saudi Tweets Dataset (STD). We wanted to understand the factors that impact the model predictions, so we used LIME (Local Interpretable Model-Agnostic Explanations), which is a tool that generates an explanation of the model prediction[34].

LIME evaluates the keywords and phrases that had an essential impact on the model's decision to classify the tweets as dialect or non-dialect. This evaluation analyzes these words and phrases that affected the model prediction. Figure 6 shows examples of the LIME explanation in the tweets by highlighting the contributed words or phrases in the model prediction and the scores representing each word's importance in the prediction. In the first example, the original text (tweet) is in Arabic: ***(يا واد قوووم)*** ; when we translated it to English, it would be *(Hey man, get up)* in Saudi dialect *(Stop lying)* and the predicted class is dialect. In the second example, the original text (tweet) is in Arabic: ***(وفرعها في السماء سؤال للجميع ؟ ماهي الشجره إلى أصلها ثابت)*** which when we translated it to English and Saudi dialect would be *(A question for everyone? What is the tree that has a firm root and its branches in the sky?)* and the predicted class is not dialect.

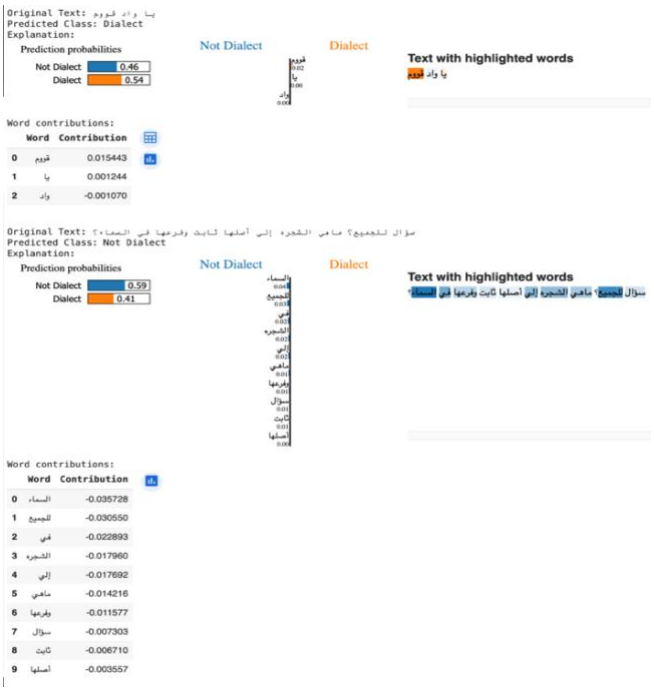

Figure 6 LIME Explanation of MARBERT Model Prediction

B. AraGPT Model Performance

During this fine-tuning process, hyperparameters and techniques were applied to assess the quality of the generated data, as the training process included four epochs, each with a batch size of 16. The learning rate was set at 5e-5, and the regularization technique to prevent overfitting in the mode was 0.01. The next step was initializing the trainer model and the generation command with 128 max length.

In order to control the balance between creativity and coherence in the generated text, we applied the following parameters. We set the temperature range to 0.7, which controls the generated text of the model, where a high value (e.g., 1.0) results in more random and diverse, and a lower value (e.g., 0.5) makes the output more deterministic. To prevent unusable vocabulary, we set top_k=50 where lower values make the model increase the focus. Also, to maintain diversity and avoid unlikely words, we put the nucleus sampling top_p =0.9. To encourage the model to generate more diverse outputs, we set repetition_penalty to 1.2, which prevents the model from repeating the exact words or phrases too often.

After that, we evaluate the AraGPT2 model performance on the perplexity metric and BLEU scores[31], [32]. On the perplexity metric, we measured the average value of the model and how effective the new tweets were[31]. A lower value means the model performs well and is more confident in its generated tweet.

The average perplexity value of the AraGPT2 model was 56.7, which indicates that the model can accurately predict the next word in the Saudi dialect. Thus, the model learned the patterns of the Saudi dialect; there is still a possibility of enhancement since the model faces some difficulty while predicting the Saudi dialect tweets.

We measured the similarity average score between generating new and reference tweets on the BLEU scores[32]. The scale of the BLEU score is from 0 to 1; a higher score means the model generates a coherent tweet that

matches the referenced tweet. In our experiment, the average score of the BLEU was 0.65, which means that the model can develop new, somewhat accurate and coherent tweets in the Saudi dialect. Hence, the generated tweets were similar to the reference tweets, yet there was some variation, particularly in mimicking the Saudi dialect. To improve these scores, further optimization is needed to enhance the text generation quality.

C. AraBERT Performance on Sentiment Analysis

The main goal of applying the AraBERT model in our research is to evaluate and understand the impact of each dataset on the model's performance on the sentiment analysis of Arabic text, specifically tweets written in the Saudi dialect. In this experiment, we focused on evaluating the model's performance on the AraCust dataset, which serves as the baseline for comparing the impact of the additional datasets.

Based on the results in Table 6 the generated Saudi dialect dataset significantly enhanced the AraBERT model performance when combined with other datasets on the sentiment analysis task. On the other hand, the model had difficulty detecting negative and positive sentiments when it was applied only to the AraCust dataset. In comparison, the model performs well when the generated dataset is added to the AraCust dataset. The generated data provides more diverse, straightforward sentences that improve the AraBERT model learning efficacy. As a result, the model accuracy increases from 89% to 97%. Also, the model demonstrates balanced performance across both positive and negative sentiments, suggesting its suitability for sentiment analysis in the Saudi dialect tweets. Furthermore, the performance of the model on only the Saudi Twitter Dataset (STD) shows the model limitations when it is fine-tuned on less comprehensive data. However, the merging of the collected dataset STD and generated data results in a significant improvement in model performance, which brings the accuracy of the model up to 95%.

Our experiment highlights how effective it is to use synthetic data to complement real-world datasets, especially in tasks where class imbalance or data scarcity could affect the model's performance. The created dataset offers the diversity and extra examples necessary to address these challenges, making it an essential factor in enhancing the performance of the AraBERT model across different datasets.

The confusion matrix in Figure 7 illustrates that the AraBERT model distinguished well between positive and negative sentiments in the Saudi dialect tweets in the combined datasets.

Table 6 Classification Report of the AraBERT model

Datasets	***Evaluation Metrics***	***Precision***	***Recall***	***F1-Score***
AraCust	***Negative***	0.98	0.88	0.93
	Positive	0.68	0.92	0.78
	Accuracy	0.89		
AraCust + Generated Dataset	***Negative***	**0.98**	**0.98**	**0.98**
	Positive	0.85	0.83	0.84
	Accuracy	**0.97**		
Saudi Twitter Dataset (STD)	***Negative***	0.79	0.65	0.71
	Positive	0.20	0.34	0.25
	Accuracy	0.58		
STD + AraCust	**Negative**	0.96	0.94	0.95
	Positive	0.90	0.93	0.92
	Accuracy	0.93		
STD + Generated Dataset	***Negative***	0.96	0.96	0.96
	Positive	**0.94**	0.92	**0.93**
	Accuracy	0.95		

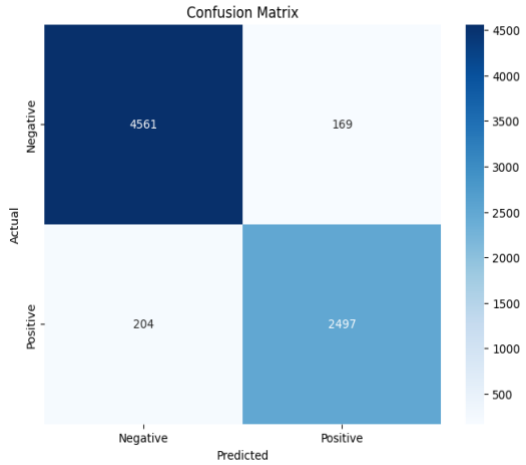

Figure 7 AraBERT Confusion Matrix

V. CONCLUSION

Due to data scarcity, sentiment analysis in low-resource languages, particularly Arabic dialects, remains challenging. However, our proposed approach demonstrates that combining generative AI with minimal human-collected data can effectively address these challenges. By fine-tuning AraBERT with a dataset consisting of collected data and a GPT-generated dataset, we achieved a notable increase in accuracy and F1 scores across all sentiment classes, with F1 scores reaching over 98% in performance metrics. Our key contributions are (1) introducing a novel framework that leverages generative AI to expand dialect-specific datasets, providing a low-cost solution for adapting standard language models to regional dialects; (2) showing that GPT-generated data can meaningfully enhance dialectal sentiment analysis for low-resource languages, as demonstrated in SD; and (3) providing visual insights with XAI LIME, explaining model predictions and demonstrating that generated data can retain linguistic nuances essential for SD sentiment classification.

While this study focuses on the Saudi dialect, our approach has the potential to be adapted for other Arabic dialects and under-resourced languages facing similar data scarcity challenges, supporting broader applicability. Additionally, as synthetic data generation scales up, maintaining data quality and dialectal authenticity remains essential, suggesting that future work could explore automated quality assurance methods. This work highlights the potential of generative data augmentation in low-resource NLP, presenting a viable path forward for improving NLP in under-resourced dialects and languages.

VI. Acknowledgment

The National Science Foundation partly supported the research under grant No. 2103563.

VII. References

[1] Audrey Azoulay, "United Nations Educational, Scientific and Cultural Organization (UNESCO)."

[2] K. of S. Arabia. S. C. General Authority for Statistics, "General Authority for Statistics, Kingdom of Saudi Arabia. Saudi Census 2020," 2020. Accessed: Sep. 03, 2024. [Online]. Available: https://database.stats.gov.sa/home/report/3868

[3] O. F. Zaidan and C. Callison-Burch, "Arabic dialect identification," *Computational Linguistics*, vol. 40, no. 1, pp. 171–202, 2014.

[4] N. Habash, O. Rambow, M. Diab, and R. Kanjawi-Faraj, "Guidelines for annotation of Arabic dialectness," in *Workshop on Arabic and its local languages*, 2008.

[5] T. Bent, E. Atagi, A. Akbik, and E. Bonifield, "Classification of regional dialects, international dialects, and nonnative accents," *J Phon*, vol. 58, pp. 104–117, 2016.

[6] L. H. Baniata and S. Kang, "Transformer Text Classification Model for Arabic Dialects That Utilizes Inductive Transfer," *Mathematics*, vol. 11, no. 24, p. 4960, 2023.

[7] A. A. Al Shamsi and S. Abdallah, "A systematic review for sentiment analysis of arabic dialect texts researches," in *Proceedings of International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2021 Volume 2*, Springer, 2022, pp. 291–309.

[8] M. El-Masri, N. Altrabsheh, H. Mansour, and A. Ramsay, "A web-based tool for Arabic sentiment analysis," *Procedia Comput Sci*, vol. 117, pp. 38–45, 2017.

[9] L. Lulu and A. Elnagar, "Automatic Arabic dialect classification using deep learning models," *Procedia Comput Sci*, vol. 142, pp. 262–269, 2018.

[10] N. Elhassan *et al.*, "Arabic Sentiment Analysis Based on Word Embeddings and Deep Learning," *Computers*, vol. 12, no. 6, 2023, doi: 10.3390/computers12060126.

[11] M. Abdul-Mageed, A. R. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," in *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2021. doi: 10.18653/v1/2021.acl-long.551.

[12] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," *arXiv preprint arXiv:2003.00104*, 2020.

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

[14] L. M. Alhazmi and A. A. Alfaifi, "Dialect identification in Saudi dialects: A socio-phonetic approach," *Journal of Language and Linguistic Studies*, vol. 18, no. 1, pp. 820–835, 2022.

[15] A. Al-Thubaity, Q. Alqahtani, and A. Aljandal, "Sentiment lexicon for sentiment analysis of Saudi dialect tweets," *Procedia Comput Sci*, vol. 142, pp. 301–307, 2018.

[16] R. M. Alahmary, H. Z. Al-Dossari, and A. Z. Emam, "Sentiment analysis of saudi dialect using deep learning techniques," in *ICEIC 2019 - International Conference on Electronics, Information, and Communication*, 2019. doi: 10.23919/ELINFOCOM.2019.8706408.

[17] L. Almuqren and A. Cristea, "AraCust: a Saudi Telecom Tweets corpus for sentiment analysis," *PeerJ Comput Sci*, vol. 7, p. e510, May 2021, doi: 10.7717/peerj-cs.510.

[18] S. Aftan and H. Shah, "Using the AraBERT Model for Customer Satisfaction Classification of Telecom Sectors in Saudi Arabia," *Brain Sci*, vol. 13, no. 1, 2023, doi: 10.3390/brainsci13010147.

[19] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2870052.

[20] M. Atabuzzaman, M. Shajalal, M. B. Baby, and A. Boden, "Arabic Sentiment Analysis with Noisy Deep Explainable Model," in *ACM International Conference Proceeding Series*, 2023. doi: 10.1145/3639233.3639241.

[21] Y. Abdelwahab, M. Kholief, and A. A. H. Sedky, "Justifying Arabic Text Sentiment Analysis Using Explainable AI (XAI): LASIK Surgeries Case Study," *Information (Switzerland)*, vol. 13, no. 11, 2022, doi: 10.3390/info13110536.

[22] A. A. Al Shamsi and S. Abdallah, "A Systematic Review for Sentiment Analysis of Arabic Dialect Texts Researches," in *Lecture Notes in Networks and Systems*, 2022. doi: 10.1007/978-3-030-85990-9_25.

[23] M. El-Masri, N. Altrabsheh, H. Mansour, and A. Ramsay, "A web-based tool for Arabic sentiment analysis," in *Procedia Computer Science*, 2017. doi: 10.1016/j.procs.2017.10.092.

[24] S. and T. C. The Saudi Communications, "2023 Report of the Saudi Internet," 2023. Accessed: Jul. 06, 2024. [Online]. Available: https://www.cst.gov.sa/en/mediacenter/pressreleases/Pages/2024042402.aspx

[25] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP," in *Procedia Computer Science*, 2017. doi: 10.1016/j.procs.2017.10.117.

[26] R. M. K. Saeed, S. Rady, and T. F. Gharib, "An ensemble approach for spam detection in Arabic opinion texts," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 1, 2022, doi: 10.1016/j.jksuci.2019.10.002.

[27] M. Abdul-Mageed, A. R. Elmadany, and E. M. B. Nagoudi, “ARBERT & MARBERT: Deep bidirectional transformers for Arabic,” in *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2021. doi: 10.18653/v1/2021.acl-long.551.

[28] A. Radford, K. Narasimhan, and I. Sutskever, “GPT: Improving Language Understanding by Generative Pre-Training,” *OpenAI*, 2018.

[29] W. Antoun, F. Baly, and H. Hajj, “ARAGPT2: Pre-Trained Transformer for Arabic Language Generation,” in *WANLP 2021 - 6th Arabic Natural Language Processing Workshop, Proceedings of the Workshop*, 2021.

[30] S. Raschka, *Build a Large Language Model (From Scratch)*. in From Scratch. Manning, 2024. [Online]. Available: https://books.google.com/books?id=sqG00AEACAAJ

[31] K. Arora and A. Rangarajan, “Contrastive Entropy: A new evaluation metric for unnormalized language models,” *arXiv preprint arXiv:1601.00248*, 2016.

[32] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002.

[33] A. Adadi and M. Berrada, “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2870052.

[34] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘ Why should i trust you?’ Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.