


Article

Enhancing Arabic Dialect Detection on Social Media: A Hybrid Model with an Attention Mechanism

Wael M. S. Yafooz 

Computer Science Department, College of Computer Science and Engineering, Taibah University,
Medina 42353, Saudi Arabia; wyafooz@taibahu.edu.sa

Abstract: Recently, the widespread use of social media and easy access to the Internet have brought about a significant transformation in the type of textual data available on the Web. This change is particularly evident in Arabic language usage, as the growing number of users from diverse domains has led to a considerable influx of Arabic text in various dialects, each characterized by differences in morphology, syntax, vocabulary, and pronunciation. Consequently, researchers in language recognition and natural language processing have become increasingly interested in identifying Arabic dialects. Numerous methods have been proposed to recognize this informal data, owing to its crucial implications for several applications, such as sentiment analysis, topic modeling, text summarization, and machine translation. However, Arabic dialect identification is a significant challenge due to the vast diversity of the Arabic language in its dialects. This study introduces a novel hybrid machine and deep learning model, incorporating an attention mechanism for detecting and classifying Arabic dialects. Several experiments were conducted using a novel dataset that collected information from user-generated comments from Twitter of Arabic dialects, namely, Egyptian, Gulf, Jordanian, and Yemeni, to evaluate the effectiveness of the proposed model. The dataset comprises 34,905 rows extracted from Twitter, representing an unbalanced data distribution. The data annotation was performed by native speakers proficient in each dialect. The results demonstrate that the proposed model outperforms the performance of long short-term memory, bidirectional long short-term memory, and logistic regression models in dialect classification using different word representations as follows: term frequency-inverse document frequency, Word2Vec, and global vector for word representation.



Citation: Yafooz, W.M.S. Enhancing Arabic Dialect Detection on Social Media: A Hybrid Model with an Attention Mechanism. *Information* **2024**, *15*, 316. <https://doi.org/10.3390/info15060316>

Academic Editor: Arkaitz Zubiaga

Received: 5 March 2024

Revised: 23 April 2024

Accepted: 27 May 2024

Published: 28 May 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Arabic text identification; LSTM; BiLSM; deep learning; social media

1. Introduction

The number of Arabic Internet users has been steadily rising, indicating an increasing digital presence in Arab speaking nations. With a population exceeding 420 million, the Arab region has witnessed advancements in Internet accessibility, mobile technology, and online services, resulting in a surge in Arabic users [1,2]. This growth can be linked to the availability of Arabic content on online platforms such as social media, e-commerce sites, and platforms for content consumption. Dealing with Arabic text poses challenges due to its varied characteristics encompassing morphology, syntax, vocabulary, and pronunciation in various dialects. These varieties include modern standard Arabic (MSA), used in formal settings, and non-standardized regional dialects, which are used and spoken in everyday communication, such as the Levantine dialect, Gulf dialect, Maghrebi dialect, etc. This distinction presents obstacles for researchers and developers in computational linguistics tasks [3,4]. Social media platforms offer insight into the diverse Arabic-speaking community by showcasing various dialects and linguistic variations. Therefore, Arabic dialect identification (AID) is a specialized natural language processing (NLP) task that is made in order to determine the specific Arabic dialect used in a given text. It is crucial for

various natural language processing (NLP) applications, including machine translation, text-to-speech synthesis, and cross-language text generation.

Many attempts have been proposed in the area of automatic dialect identification (ADI), and early uses are based on dictionaries, rules, and language modeling [5–10]; more recently, a shift was made toward employing machine learning techniques [11–24], deep learning approaches [25–40], and transfer learning methods [41–49]. Many of these investigations utilize prominent and accessible datasets, such as MADAR [49], NADI [50–53], QADI [54], MPCA [55], and Dart [56]. Furthermore, explorations into the challenges and concerns of ADI have been explored in [57], with additional, comprehensive analyses provided in [58–61]. Machine learning techniques are used increasingly to classify Arabic text, enabling computers to categorize and understand the content of textual data written in Arabic automatically. For Arabic text categorization problems, deep learning models like convolutional neural networks (CNN) and recurrent neural networks (RNN), as well as algorithms like support vector machines (SVM), naive Bayes (NB), and decision trees (DT), are frequently used.

To improve the performance of the models, researchers may apply techniques like word embeddings, such as global vectors for word representation (GloVe), Word2Vec, and term frequency-inverse document frequency (TF-IDF), which capture semantic relationships between words in the Arabic language [62]. Moreover, models that fall into the category of pre-trained language (LMs), such as bidirectional encoder representations from transformers (BERT), were adapted in order to handle Arabic text, allowing for more accurate and context-aware classifications. The success of machine learning in classifying Arabic text heavily relies on having a diverse and well-labeled dataset. Moreover, continuous research and development in the field are crucial to deal with the issues raised by the complexities and nuances of the Arabic language, such as dialectal variations, code-switching, morphological complexity, limited resources, and the lack of standardized corpora. Thus, this study presents a novel hybrid deep learning model for identifying and classifying Arabic dialects on written language lexically. The model is composed of long short-term memory (LSTM), bidirectional sequential memory (BiLSM), and logistic regression (LR) components. Several experiments were conducted using an introduced dataset. This dataset was gathered from the Twitter platform and encompassed 34,905 entries distributed across the following four imbalanced classes: Egyptian, Gulf, Jordanian, and Yemeni dialects. The outcomes of the experiments highlight that the proposed model surpasses the performance of existing baseline models, establishing its effectiveness in Arabic DID in terms of accuracy. The contributions of this study can be viewed from fifth folds, as follows:

- A novel hybrid machine and deep learning model consisting of LSTM, BiLSTM, and logistic regression are proposed. This model is designed specifically for detecting and classifying Arabic dialects;
- A new dataset comprising four Arabic dialects, namely, Egyptian, Gulf, Jordanian, and Yemeni, is introduced. This dataset is likely collected and curated for training and evaluating the proposed model;
- The performance of the proposed model is examined. It comprises training the model on the novel dataset and evaluating its accuracy, precision, recall, F1-score, or other relevant metrics to measure its ability to identify and classify the different Arabic dialects;
- The proposed model's performance is examined using different word representations, namely TF-IDF, Word2Vec, and GloVe, on the introduced dataset for the Arabic dialect.

The remaining part of this paper comprises several sections. Section 2 provides a literature review related to existing approaches for Arabic text identification and dialect classification. The problem formulation of Arabic dialect identification explained in Section 3. Section 4 details the methodology and materials used, including the novel dataset of the following four Arabic dialects: Egyptian, Gulf, Jordanian, and Yemeni. The results and discussion, showcasing the performance of the model and providing an in-depth analysis of its strengths and limitations, is presented in Section 5. Finally, the last section concludes this paper, summarizing its contributions and suggesting potential future research in this field.

2. Related Studies

This section studies the concern of dialect text identifications. It has gained significant attention owing to the prevalence of dialectal variations in many languages, including Arabic, English, and Spanish. Researchers have explored various techniques to improve dialect text identification accuracy. This study categorized the techniques into the following four categories: machine learning, deep learning, transform learning, and language modeling.

Traditional machine learning methods, such as SVM, NB, and DT, have been employed for dialect text classification. These models often rely on hand-engineered features extracted from the text, such as lexical, syntactic, and morphological features, to distinguish between dialects. Ali et al. [11] developed a way to differentiate between Arabic dialects distinctly by taking bottleneck characteristics from the i-vector framework and phonetic and lexical data from a speech recognition system, which are then used to identify dialects in Arabic broadcast speech. Similarly, Boujou et al. [12] introduced an open data collection of social data material in Arabic dialects obtained from the Twitter social network. The researchers then evaluated the dataset using four different classifiers, namely SGD, LR, NB, and linear SVC, with NB achieving the highest accuracy of 0.79%. Sobhy et al. [13] used language to predict social media user dialect through country level and sentiment analysis. El-Haj et al. [14] developed a subtractive bivalency profiling technique and expanded on prior work by including new grammatical and stylistic criteria. Butnaru and Ionescu [16] developed a method for categorizing Arabic dialects, combining a kernel ridge regression classifier with multiple-kernel learning. Their approach incorporates kernels, such as character n-grams, dialectal embeddings, and kernel discriminant analysis. Johnso et al. [17] explored automated dialect density prediction in African American English (AAE) speech using X-vector characteristics and deep learning methods. They utilized audio segments from the CORAAL database for their study. Hassani and Dzejla [18] created a classifier to detect different dialects in Kurdish, which has a variety of dialects and no established orthography. The technique may be used with different Kurdish dialects and languages that have many of the same dialectal characteristics as Kurdish.

With the advent of deep learning, and more specifically CNNs and RNNs, dialect text classification has advanced significantly in recent years. Deep learning models are successful because they can extract complex patterns from raw text input, which is crucial when dealing with dialectal variances. Semantic word embeddings like Word2Vec and GloVe have been shown to be useful in capturing the semantic relationships between words, with the overall aim of improving dialect text classifiers. As previous LMs that are not pre-trained, such as BERT, have achieved the best scores in numerous languages, Mohammad et al. [25] fed LSTM with dialectal semantics and achieved excellent results. Sundus et al. [26] proposed a feed-forward DL neural network that uses TF-IDF vectors to classify Arabic text with lower classification error rates. Alqureshi [27] devised an Arabic dialect classifier for more than 400 million Arabic speakers with specific attention to fine-grained Arabic dialects, such as the one in Saudi Arabia. Abdelazim et al. [28] came up with a suggested CNRN-RNN for the Arabic DID problem, which mainly included Egyptian, Levantine, or Gulf dialects dialect classifiers. Fares et al. [29] offered methods for dialect determination by means of frequency-related features and classifiers without deep learning on the MADAR small-scale corpus. ELaraby and Abdul-Mageed [30] employed deep learning-based dialect recognition through the use of a separate classifier that provides a solid job baseline, as well. El Mekki et al. [31] suggested an end-to-end deep multi-task learning system that mixed the task-discriminative and inter-task characters shared across the tasks. Wang et al. [32] explored the application of LSTM and word embeddings in sentiment classification on social media, taking into account lengthy and variable content. Nowak et al. [33] observed the use of LSTM networks and their versions for sentiment and short text classification with the aim of dealing with the vanishing gradient problem in RNNs and comparing LSTM, bidirectional LSTM, and gated recurrent unit (GRU) networks.

Transfer learning can significantly reduce the need for extensive training data and computational resources. It makes the model's performance more efficient and accurate

than training from scratch. Abdul-Mageed et al. [41] proposed a deep learning model that removes irrelevant dialect information and constructs a vector representation of the most relevant token. The LSTM model and transformer model are applied using BERT. Three datasets, namely, MADAR, NADI, and QADI, have been utilized based on the multiclass problem. Abdelali et al. [42] utilized AraBERT and BERT with fine tuning using the proposed dataset called QADI and the publicly available dataset MADAR. QADI automatically collected Tweets from 18 distinct Middle Eastern and North African countries representing diverse country-level Arabic dialects of 18 Arab nations. In Alghamdi et al. [43], three distinct classification tasks were completed (with all class labels accessible in each dataset) as follows: binary dialect classification, three-way dialect classification, and multi-way dialect classification. Attieh and Hassan et al. [44] developed two deep learning approaches based on AraBERT for nuanced Arabic dialect identification (NADI). In the same way, Fsih et al. [45] and Messaoudi et al. [46] participated in the NADI competition. However, Messaoudi (Compass team) used a MARABERT V2 model, which gave him the highest score from the findings of the research, which was 0.50, and reached the fourth-highest accuracy in terms of test-A for subtask 1 in the competition (51.91). Fsih et al. [45] used three models for NADI 2022 (sentence transformer, CAMElBERT, and multi-dialect BERT). Joseph Attieh and Fadi Hassan from Finland, in the first subtask, distinctly divided between Western (Morocco, Algeria, Tunisia, and Libya) and Eastern Arabic dialects (Egyptian, Levantine, and Gulf) with a dataset of over 20,000 Tweets with the AraBERT model, and the highest score received was 74.64%.

Language modeling in Arabic identification refers to utilizing computational methods to comprehend the distinct features of the Arabic language. By constructing advanced models, this approach captures Arabic text's intricate grammar, semantics, and patterns, enabling the accurate interpretation of written or spoken content. Baimukan et al. [63] mapped 29 diverse datasets to a uniform three-level hierarchical paradigm for DA categorization. These datasets may be combined more easily given the common schema; with all the datasets they worked with, they constructed aggregated n-gram LM region, nation, and city levels in character and word spaces. Etman and Beex [58] investigated prosodic and phonotactic data in Arabic-specific artificial dialect recognition systems focusing on extra-linguistic features. The identification accuracy of prosodic elements increases, making ADI a difficult problem in speech and language recognition. Huang [7] created a method for improving Arabic dialect categorization using semi-supervised learning. Accuracy is significantly increased by training several classifiers on weakly supervised, strongly supervised, and unstructured data. Obeid et al. [64] introduced Camelira, an online tool for multi-dialect Arabic morphological disambiguation that caters to MSA, Egyptian, Gulf, and Levantine, the most prevalent language dialects. Camelira can automatically select an appropriate dialect-specific disambiguator using a DID component. An analysis of existing ADI methods is presented in Table 1.

Table 1. Comparative analysis between existing methods of ADI.

Author(s)	Techniques	Dataset Size	Dialect	Platform	Classifiers
[11]	Machine Learning	Voice recordings	Egyptian, Gulf, Levantine, and North African.	Al-Jazeera	SVM
[12]	Machine Learning	50k Tweets	Algeria, Egypt, Lebanon, Tunisia and Morocco.	Twitter	SGD Classifier
[13]	Machine Learning	20K Tweets		Twitter	TF/IDF, MNB, CNB, SVM, KNN, DT, RF, MLP
[14]	Machine Learning	16,494 Sentences	Egypt, North Africa, Gulf and Levant, MSA	AOC dataset	KNN, NB, SVM

Table 1. Cont.

Author(s)	Techniques	Dataset Size	Dialect	Platform	Classifiers
[15]	Machine learning	2000 Sentences	MSA, Egyptian, Syrian, Jordanian, Palestinian, Tunisian	Multidialectal Parallel Corpus of Arabic (MPCA)	N/A
[16]	Machine Learning	Voice Recordings	EGY, GLF, LAV, and North African or NOR	Arabic broadcast speech	KRR
[17]	Deep Learning	Audio segments	African American English	CORAAL database	XGBoost model
[18]	Machine Learning	7000 words	Sorani, Kurmanji (Kurdish Dialects)	vocabulary words	Adaptation of SVM
[31]	Deep Learning	31k Tweets	Maghreb, Egypt, Gulf, and Levant	Twitter	MTL
[25]	Deep Learning	2000 MSA sentences and 540k Tweets	Algerian, Bahraini, Djiboutian, Egyptian, Iraqi, Jordanian, Kuwaiti, Lebanese, Libyan, Mauritanian, Moroccan, Omani, Palestinian, Qatari, Saudi, Somali, Sudanese, Syrian, Tunisian, Emirati, and Yemeni.	Twitter	LSTM
[26]	Deep Learning	7135 Documents	Arabic	Khaleej-2004 Corpus Dataset and newspaper articles	feed-forward DL neural network model
[27]	Machine Learning and Deep Learning	3768 Sentences	Hijazi, Najdi, Janobi, Hasawi (Saudi Dialects)	blogs, discussion forums, and reader commentaries	SVM LR SGDC CNN
[28]	Deep learning	34 h	Egyptian Gulf and Levantine	52 Volunteer Participants	Gaussian NB, SVM, RNN and CNN-RNN
[41]	Transform Learning	10 M Tweets	Egyptian, Iraqi, Jordanian, Saudi, Kuwaiti, Omani, Palestinian, Qatar, UAE, Yemen	Twitter	MARBERT
[42]	Transform	540k Tweets	Emirati, Bahraini, Djiboutian, Egyptian, Iraqi, Jordanian, Kuwaiti, Lebanese, Libyan, Mauritanian, Omani, Palestinian, Qatari, Saudi, Sudanese, Syrian, Tunisian, and Yemeni.	Twitter	AraBERT
[43]	Transform and deep learning	100k Records 1.4 M Records	Gulf, Iraqi, Egyptian, Levantine, and North Africa dialects.	AOC dataset, SMADC dataset	MARBERT, ARBERT
[46]	Deep Learning and transform learning	25,269 Sentences	Arab World	NADI	Arabic BERT MARBER
[58]	Language model/feature extraction	6300 sentences of speech	American English Dialects (Southern and South midlands)	Voice Recordings	backend logistic classifier

Table 1. Cont.

Author(s)	Techniques	Dataset Size	Dialect	Platform	Classifiers
[63]	Hierarchical aggregation	N/A	Levantine, Gulf, Iraqi, Omani, Egyptian, North African, Yemeni	Twitter and voice speech MADAR	LM
[65]	Statistical analysis	36 Human Speakers	Chungli, Changki, Mongsen (Nagaland)	Speech and Written text for Chungli dialect only	GMM
[7]	semi-supervised learning	11.8k Sentences	Egyptian, Gulf, Iraqi, Levantine, Maghrebi	AOC Corpus and Facebook	N/A
[66]	Semi/Unsupervised	3000 h of speech	Algerian, Egyptian, Iraqi, Jordanian, Saudi, Kuwaiti, Lebanese, Libyan, Mauritanian, Moroccan, Omani, Palestinian, Qatari, Sudanese, Syrian, Emirati, Yemeni	YouTube	N/A

3. Problem Formulation of Arabic Dialect Identification

This section explains the problem formulation for Arabic text identification. The primary challenge lies in the domain of natural language processing, specifically in accurately identifying and categorizing Arabic text, which falls within the domain of natural language processing. To address this, we compiled Arabic text samples from user-generated comments on Twitter, encompassing the following four distinct Arabic dialects: Egypt, Jordan, Gulf, and Yemeni. As a result, this forms a multiclass problem, in which the dataset of user/Twitter comments is referred to as $\{t\}$, and the classes representing the Arabic dialects are denoted by $\{AC\}$.

The given problem involves a set of shared posts on social networking platforms, specifically Tweets, denoted as $Tw = \{t_1, t_2, t_3, \dots, t_n\}$. Each $\{t\}$ in the dataset represents a user comment/Tweet, $t \in Tw$, and $\{n\}$ denotes the total number of Tweets in the set.

The task at hand involves classifying each Tweet in the set Tw into specific classes represented by $AC = \{C_1, C_2, C_3, C_4\}$, where $C \in AC$. These classes correspond to distinct Arabic dialects, where C_1 represents the Egyptian dialect, C_2 represents the Jordanian dialect, C_3 represents the Gulf dialect, and C_4 represents the Yemeni dialect.

The classification process aims to associate each Tweet with the appropriate dialect class based on its linguistic characteristics and context. By utilizing Formula (2), the goal is to develop a classification model that can accurately detect and categorize Tweets into one of the four defined dialect classes.

The dataset is denoted as $\{D\}$, and it can be split into two parts, $\{D_1\}$ and $\{D_2\}$, for testing and training the proposed model, respectively, where $D = D_1 \cup D_2$. During the training phase, the proposed model is trained using the extracted features from the set of shared posts on social networking platforms (Tweets), denoted as T . Each Tweet in T is represented as a numerical vector, V , which serves as the input to the machine learning or deep learning algorithms used for training. The numerical vector V is created based on the specific features extracted from the Tweets and is designed to capture the relevant information needed for classification. The set of numerical vectors representing features of each Tweet in T is denoted as $V = \{V_1, V_2, V_2, \dots, V_k\}$, where k is the total number of user-generated comments (Tweets) in T . Each V is represented in the $TF - IDF$ representation.

The training process was conducted using D_1 . Throughout the testing phase, the model takes the numerical vector representation, V , of each Tweet which is represented by $TF - IDF$ and applies a function, denoted as F , to measure and determine which class C the Tweet belongs to. Class C corresponds to one of the predefined categories

representing the different Arabic dialects, such as Egyptian, Jordanian, Gulf, or Yemeni dialects as $T \rightarrow V(TF - IDF)$ which is converted as input X and is classified to one of the classes, $F(TF - IDF) \in C$, and F represents the activation function and formulizes in the matrix in Figure 1. Figure 1 shows that the X matrix represents the input matrix, and the first row in the matrix shows the bias. While W in the matrix represents the weight, the formulation of X and W is shown to give y in Equation (1). Let y represent the logits which are known as values before the activation function F . Then, the Softmax activation computes the probability \hat{y} to assign it to the class i (C_1, C_2, C_3 , and C_4), as shown in mathematical Formula (2), where K is the total number of classes.

$$y = X \times W \tag{1}$$

$$\hat{y} = \frac{e^{y_i}}{\sum_{j=1}^k e^{y_j}} \tag{2}$$

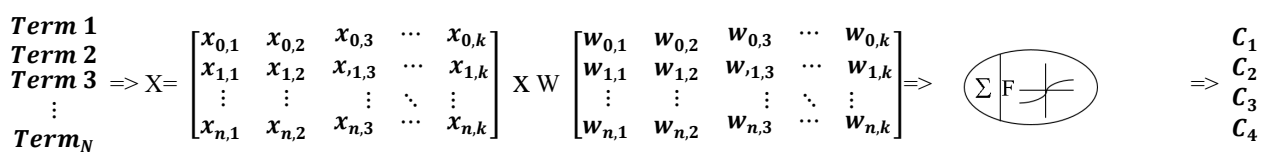


Figure 1. Matrix representation.

The objective of this process is to calculate and assess the model’s ability in order to accurately classify Tweets into their respective dialect classes during testing, based on what it learned during the training phase using D_1 . By achieving high accuracy and performance on D_1 (testing set), the proposed model demonstrates its effectiveness in classifying Arabic text and identifying the dialects of user-generated comments on social networking platforms.

4. Methods and Materials

This section identifies the various phases and methods used to conduct the research on Arabic text identification. As shown in Figure 2, the suggested paradigm for identifying and categorizing Arabic text consists of multiple phases. The ensuing subsections contain a detailed explanation of each step.

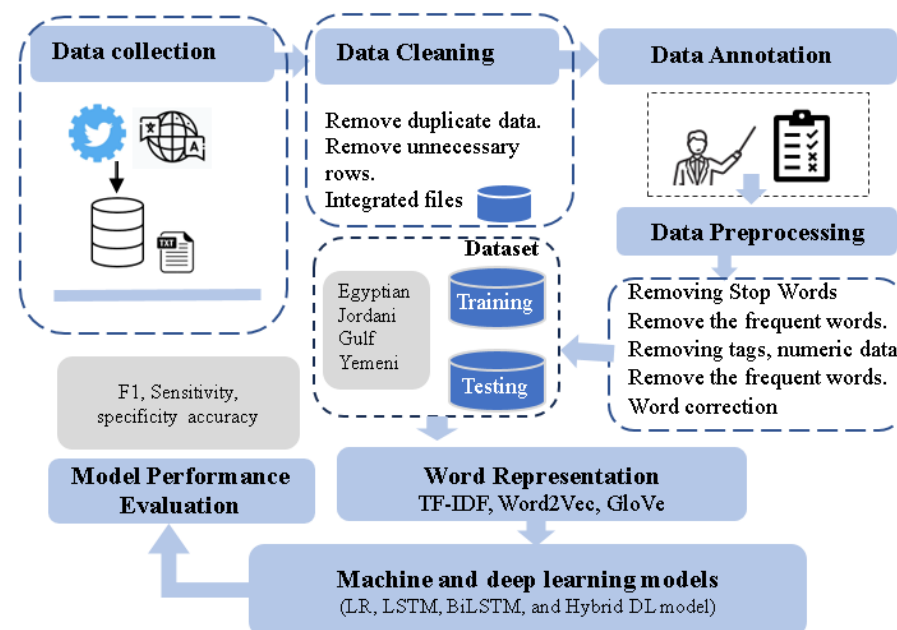


Figure 2. Model architecture.

4.1. Data Collection Phase

In the first phase of the research, the focus is on gathering pertinent data from social media platforms, specifically user-generated comments sourced from Twitter. This data collection process was carried out utilizing two libraries, namely, Tweepy and sncrape. In using Tweepy, a Twitter developer account was created in order to obtain the necessary keys (API key, secret, access token, and secret). These keys were employed in the Python programming language, along with specific parameters like the search query to access user comments. The extracted data was then stored in a CSV file. Utilizing sncrape, “sncrape.modules.twitter” was employed to interact with the Twitter API to extract user comments and save them into an Excel file using search parameters. The goal was to accumulate a significant amount of user comments from Twitter to form a substantial dataset. The primary objective was to construct a comprehensive dataset that encompasses a diverse range of Arabic dialects, ensuring ample variation in the collected text samples. The main criteria for collecting the data were user-generated comments from Twitter, a comment period between July 2022 and July 2023 and the Egyptian, Gulf, Jordanian, and Yemeni dialects. To achieve this, data were gathered from a wide array of public figures and users with diverse backgrounds, including politicians, actors, actresses, sports fans, and more. The comments, along with their associated metadata, were extracted and consolidated into a comma-separated value (CSV) format from the Twitter platform. The resulting CSV file includes various metadata descriptions, as detailed in Table 2. The content item—which holds the textual data derived from user comments—remains after all non-essential elements have been eliminated. A total of 113,207 comments, collected from different mobile delivery apps, make up the dataset.

Table 2. Dataset description.

Dialect	Size	Min (Words)	Max (Words)
Egypt	9461	1	174
Jorden	7705	1	496
Yemen	7238	1	42
Gulf	10,501	1	74
Total	34,905		

4.2. Data Cleaning

This phase comprises the following two primary components: data cleaning and data pre-processing. These components are crucial in preparing the data for the subsequent phases of the research. During data cleaning, various essential steps have been executed to ensure the dataset’s quality and relevance. Irrelevant columns have been removed to streamline the dataset, retaining only pertinent information. Duplicate data have been eliminated to avoid redundancy and ensure data integrity. Moreover, any data noise, such as irrelevant or inaccurate information, has been filtered out to enhance the dataset’s overall quality. Additionally, specific data-cleaning techniques have been implemented to address instances in which Arabic sentences were written using English letters, despite their intended meaning being in Arabic. These instances have been rectified to accurately represent the intended Arabic language content within the dataset.

4.3. Data Annotation

In the annotation process, three Arabic native speakers participated in assigning user comments to Arabic dialect classes, as defined previously. Each class was annotated by these three annotators. Once the annotation was completed, the agreement between the annotators for the classes was verified using Cohen’s Kappa measure, which demonstrated a strong agreement of 81%. The resulting dataset, after the completion of the annotation and verification tasks. This dataset serves as the foundation for further analysis and model development in this study. An example of an Arabic dialect is shown in Table 3.

Table 3. Example of Arabic dialect in proposed dataset.

Dialect	Example in Arabic	Example in English
Egypt	يا عم ازيك و انت ليه اساسا تكلمني انجليزي و احنا مصريين زي بعض	Uncle, how are you, and why do you even speak English to me when we are Egyptians like each other?
Jordan	يا زلمة والله إنك ولد	Oh man, by God, you are a boy
Yemen	اشتی اعرف اتتوا فين بتروح اليوم	I want to know where you go today
Gulf	وايش تبغا منه	What do you want from him?

4.4. Pre-Processing

The dataset, which consists of noisy, unstructured text data, is handled at the data pre-processing stage in order to possibly affect the model's performance. To enhance the accuracy of classifying the classes, data pre-processing techniques are applied. Common natural language processing (NLP) methods used in this pre-processing involve removing stop words, eliminating repeated characters in sentences, and discarding special characters, numbers, and punctuation using regular expressions. Additionally, stemming methods are utilized to revert Arabic words to their root forms. These steps are essential in handling unnecessary data that could otherwise hinder the model's performance. By thoroughly cleaning and refining the dataset through these pre-processing techniques, the model's capability to accurately classify the classes is significantly improved. As a result of meticulous data cleaning and pre-processing, the dataset is now primed for further analysis and the development of the model in the subsequent phases of the research.

The pre-processing steps applied to the data include the following:

- Removal of Special Characters: Special characters like “#” and “@” commonly used in Tweets are removed from the text;
- Elimination of English Words or Characters: Any English words or characters, such as mentions or references to others, are removed from the text;
- Exclusion of English Numbers: Numerical values in English are eliminated from the text;
- Exclusion of Arabic Numbers: Arabic numerical values are taken away from the text;
- Elimination of Tweets with No Words: Tweets that do not contain any words, such as those consisting only of images, mentions, or characters, are dropped from the dataset;
- Augmentation with Arabic Stop Words Removal: To enhance the analysis, the data is processed twice. The first time follows the previous steps, and the second time involves an additional step of removing Arabic stop words. The stop words are extracted from the NLTK Python libraries and include words that do not significantly contribute to sentimental analysis or dialect classification, such as “و” (and), “أو” (or), “إلا” (except), “لكن” (but), and so on.

4.5. Word Representation

During this phase, the focus is on feature extraction methods that aim to transform the textual data into numerical representations. Three primary word representation techniques, namely, TF-IDF, Word2Vec [67], and GloVe [68], are utilized to examine the model's performance using each method. TF-IDF is a popular technique used to convert textual data into numerical vectors. It captures the importance of every word in a document within a larger collection of documents. The process assigns a score to every word according to its rarity throughout the collection (inverse document frequency) and its frequency within the document (term frequency). As a result, each document's unique qualities are emphasized in a numerical vector representation.

$$TF_{t,Tw} = \frac{\text{frequency of term 't' in twitter 'TW'}}{\text{total terms in document 'Tw'}} \quad (3)$$

$$IDF_t = \log_{10} \frac{\text{total number of twittes}}{\text{total number of twittes with term 't'}} \quad (4)$$

A word embedding method called Word2Vec seeks to identify the semantic connections among words in a corpus. Words with comparable meanings are mapped to dense vectors in a continuous vector space, in which each word is mapped. This allows for the representation of words in a more meaningful and context-aware manner, which can improve the performance of machine learning models. The Word2Vec algorithm simplifies the representation of words in a continuous vector space by using either the continuous bag of words (CBOW) or Skip-gram architecture. These architectures learn word embeddings, which are dense vector representations of words. CBOW predicts the target word from its given context, while Skip-gram predicts the context of the words given to the target word. Both architectures use neural networks to optimize their training objectives and learn meaningful word embeddings that take in semantic relationships between words. These word embeddings are useful in various natural language processing tasks.

GloVe is another word embedding method that combines the advantages of Word2Vec and matrix factorization techniques. It generates word vectors by considering both the global word co-occurrence statistics and the local word context information. This approach results in word representations that exhibit better semantic relationships and capture global word meanings.

4.6. Machine and Deep Learning Models

In this study, four distinct models were employed to address the research problem. The first model utilized is logistic regression, a classical machine learning algorithm that aims to classify data into different classes based on a linear combination of input features. Logistic regression was chosen as the classifier, influenced by its promising outcomes identified in the literature review. Its suitability stems from its ability to achieve high accuracy when dealing with textual data and multiclass problems.

The second model is LSTM [69], a type of recurrent neural network (RNN) widely used for its ability to process sequential data, retain long-term dependencies, and capture contextual information. It excels in tasks involving time series analysis, natural language processing, and sequential data, as it effectively learns from variable-length input sequences and preserves hierarchical information. The LSTM's unique architecture with memory cells and gating mechanisms allows it to overcome the vanishing gradient problem and handle long sequences.

The third, BiLSTM [70], is an extension of LSTM that processes the input sequence in both forward and backward directions, enhancing the model's ability to capture contextual information. Finally, this study introduces a novel hybrid deep learning model with an attention mechanism to detect and classify the Arabic dialect, which likely combines elements from various deep learning architectures with traditional machine learning methods to leverage their individual strengths and potentially improve performance.

The proposed hybrid deep learning model comprises the following three parallel branches: LSTM with two layers, BiLSTM with two layers, and logistic regression; these are followed by a squeeze layer to adjust data input dimensions to LR, attention self-sequence layers for each branch, and a concatenate layer, followed by a flatten layer. Using the attention self-sequence layer helps the model pay more attention to certain parts of the input sequence when making predictions, instead of treating everything the same [71,72]. Therefore, it emphasizes the important words from contextual information. Dropout layers (10% and 15% rates) are integrated to mitigate overfitting, and subsequent artificial neural network layers process the concatenated data for the final output. This intricate architecture is designed to handle sequential data with attention mechanisms and prevent overfitting by introducing dropout regularization. The proposed models without and with the attention mechanism are presented in Figures 3 and 4, respectively.

Three-word representation has been utilized, namely TF-IDF, Word2Vec, and GloVe. Word2Vec was downloaded from vectors.nlp.repository/ with ID 31. The corpus name is

Arabic CoNLL17 corpus, which has a vector size of 100, a window that equals 10, and a vocabulary size of 1,071,056 and uses the Word2Vec Continuous Skip-gram algorithm. The chosen model was GloVe-twitter-100 from the gensim library, which is a pre-trained model that is trained on Twitter to return a vector size of 100.



Figure 3. Proposed model without attention mechanism.



Figure 4. Proposed model with attention mechanism.

4.7. Model Evaluation

This section outlines the evaluation methods employed to assess the model performance of the proposed hybrid deep learning model and compares its experimental results

with those of common machine and deep learning models. To achieve this, the confusion matrix is utilized to extract key performance metrics such as accuracy, precision, recall (sensitivity), and F1-score. The hybrid model is trained on a designated dataset, and predictions are generated on a separate testing dataset. By constructing the confusion matrix with predicted and actual labels, the model's performance is thoroughly analyzed, enabling a comprehensive comparison with traditional machine learning and deep learning models. The evaluation process aims to demonstrate the potential strengths and advantages of the proposed hybrid model in tackling the specific problem domain while establishing its competitiveness against established approaches. The accuracy mathematical formula is represented in Formula (5), while the *F1* score is represented in mathematical Formula (6).

$$Accuracy = \frac{TP + TN}{All} \quad (5)$$

In the accuracy formula, true positive (*TP*) is the total number of correctly classified comments that have been labeled as positive by the model, while true negative (*TN*) is the total number of correctly classified comments in which the model has labeled the comments as negative. Furthermore, "All" represents all the comments gathered pre-classification. We can see that the accuracy is gathered by taking the *TP* and *TN* comments from all the comments gathered using the mathematical equation shown in (5).

$$F1 \text{ score} = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (6)$$

The *F1* score is achieved by taking the precision and recall scores, which consist of *TP*-, *FP*-, and *FN*-classified comments, and calculating the scores using the presented equation, as seen in (6).

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

The precision is calculated by taking the *TP*-classified comments by themselves and the false positive (*FP*). *FP*-classified comments are when the model labels a comment to be positive and the label provided by the model is incorrect.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

The recall for the model can be generated by taking the *TP* comments by themselves along with the false negative (*FN*). *FN*-classified comments are when the model labels a comment to be negative and the label provided by the model is incorrect.

5. Results and Discussion

This section explains and discusses the experimental results of the proposed model for detecting and classifying Arabic dialects compared with the state-of-the-art models. Using both deep learning models like LSTM and BiLSTM and more conventional machine learning classifiers like logistics regression, the model was assessed in a number of trials. A brand new hybrid model was also unveiled. CoLab was used to conduct all of the experiments using the Python programming language. A total of 20% of the dataset is used for testing, while the remaining 80% is used for training.

The experiments involved training these models on labeled datasets containing samples of different Arabic dialects and then evaluating their performance on a separate test dataset. However, specific details about the model architectures, datasets, hyperparameters, and evaluation metrics were not provided in the concise summary, making it necessary to refer to the full study for a comprehensive understanding of the experiment environment and results. The experiment configuration is shown in Table 4. These precise parameters were determined using best practices in the field of natural language processing, numerous tests, and prior research. Multiclass classification tasks use the "Adam" [73–76] optimizer and

the “categorical_crossentropy” loss function [77,78]. A series of exploratory experiments were conducted to determine the best values for the input shape, batch size, and number of epochs in order to maximize model performance and minimize overfitting. In particular, four experiments have been conducted. The model performance in all experiments has been evaluated using accuracy, F1, precision, and recall.

Table 4. Parameters that were used in experiments.

Parameters	LSTM	BiLSTM	Proposed Model
Cost function	categorical_crossentropy		
Optimizer	adam		
Input shape	10,000 for TF-IDF and 100 for others		
Batch size	32	16	16
Epochs	70	70	50
Activation function (Hidden layer)	Relu [79]		
Activation function (Output layer)	Softmax		
Dropout	10–15%		

In the four experiments, word representation TF-IDF, Word2Vec, and GloVe have been applied in order to examine which of the methods give the highest accuracy. The first experiment was carried out using the hyperparameter and experiment configuration in Table 4. Within this experiment, the LR model was executed utilizing the following three techniques of word representation: TF-IDF, Word2Vec, and GloVe as shown in Table 5. These techniques were employed during the testing phase and validation. The observed disparities in scores exhibited a minimal deviation of no more than 5%. As a result, there are no overfitting or underfitting issues. The evaluation of the model’s performance, in terms of accuracy, was primarily based on the TF-IDF method, which recorded 83%. It is noteworthy that an accuracy rate of 77% was attained for both the Word2Vec and GloVe approaches.

Table 5. Comparison between the accuracy of three-word representation methods.

Model	TF-IDF	Word2Vec	GloVe
LR	82.95%	77.89%	77.89%
LSTM	79.76%	79.30%	79.55%
BiLSTM	80.61%	79.46%	80.79%
Hybrid	82.96%	81.67%	80.18%
Hybrid with attention	83.31%	81.31%	81.22%

In the second experiment using the LSTM, the accuracy is almost the same using three techniques of word representation with an average of 79%. While using the same experiment hyperparameters, the accuracy slightly increased, which was recorded as 81% on average, using the three techniques of word representation using BiLSTM. The proposed two models, namely, hybrid and hybrid with attention, received the highest scores in all three techniques of the three-word representation methods in the testing, training, and validation phases. This score was achieved using a few epochs, approximately 50, with the hybrid with attention mechanism outperforming the hybrid model by 2%.

However, it outperformed the hybrid model in the Word2Vec word representation method, which received 81.6%, and the hybrid model with attention, receiving a score of 81.3%. It is also to be noted that the LR model, which was the only machine learning model that was tested in the experiment, received the third highest score in all three techniques of word representation methods, and it was behind both of the proposed models (hybrid and hybrid with attention); however, beyond the LSTM and BiLSTM models, the score that the LR model reached was 82.9% through TF-IDF, 77.8% through Word2Vec, and 77.8% through GloVe. The LSTM model came last in terms of performance on word representation and was outperformed by BiLSTM, which is the fourth-highest-performing model. It is to be noted that overall, all the models performed at their highest peak when the models were

tested on the TF-IDF word representation method, for which all the models received an average score of 81.9%. Also, all the models' performance individually was at the highest extent, with the proposed models excelling in terms of performance, in the TF-IDF method, except for the BiLSTM model, which obtained the highest score in terms of being tested in the GloVe method and received a TF-IDF score of 80.6% and a GloVe score of 80.7%.

The Word2Vec method was the last in terms of performance for word representation, except for the aforementioned BiLSTM model, and both of the proposed models obtained a higher Word2Vec score of 81.6% (hybrid) and 81.3% (hybrid with attention). It should also be mentioned that the Word2Vec method achieved an overall average score of 79.9% for all the models it was tested with. Lastly, the GloVe method was the last in terms of performance, except for the LR and LSTM models, which both received a score of 77.8% (LR) and 79.5% (LSTM), both of which were their second-highest scores when being tested with the GloVe method. The confusion matrix of the hybrid model and hybrid model with attention using the three-word representation are shown in Figure 5. The performance of the proposed model in terms of accuracy and function loss of training and validation are presented in Figures 6 and 7, respectively.

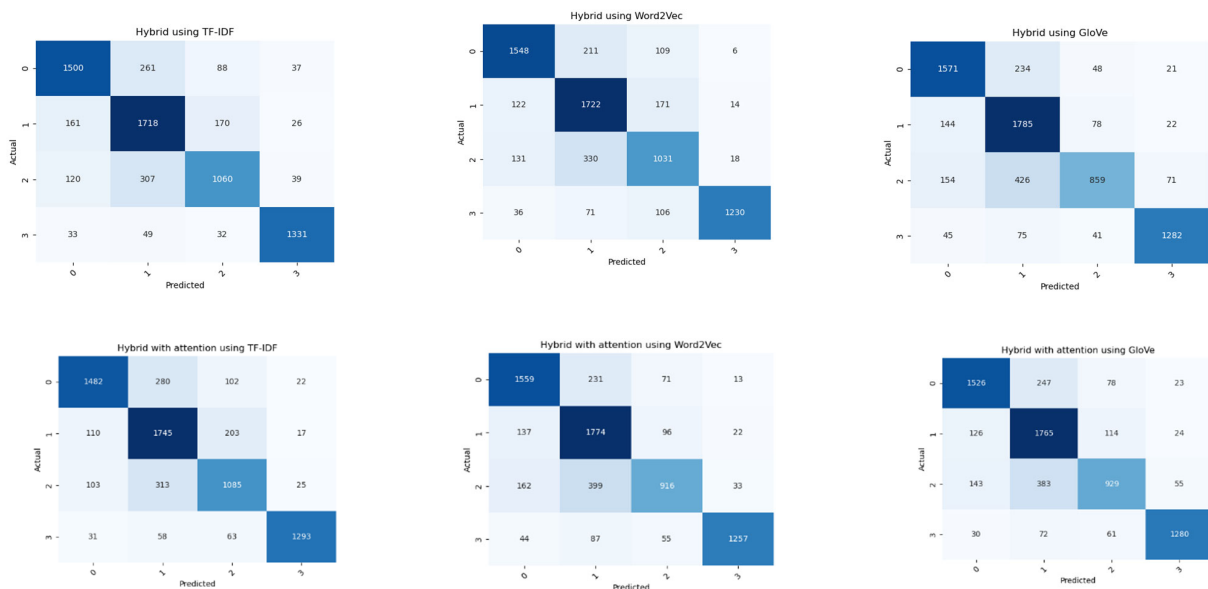


Figure 5. Confusion matrix of the proposed model.

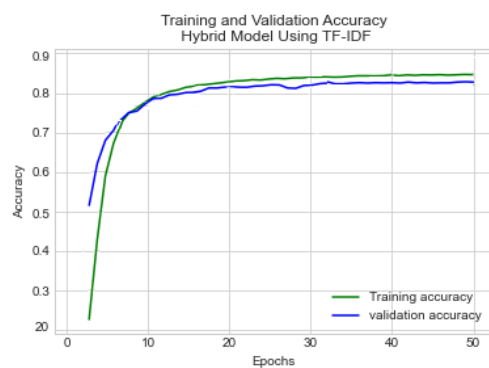


Figure 6. Accuracy of the proposed model with attention mechanism.

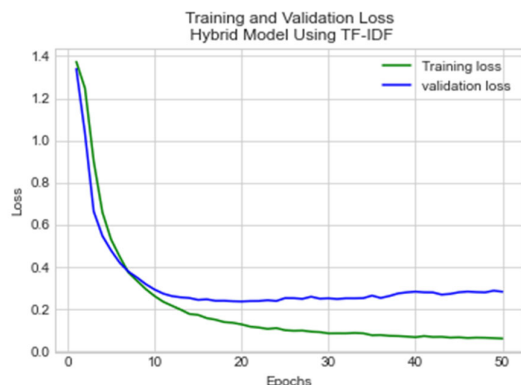


Figure 7. Loss of training and validation.

With all this in mind, it is worth mentioning that the proposed models were the best models tested in terms of word representation in all three methods tested (TF-IDF, Word2Vec, and GloVe). Both achieved the highest scores when being tested with the TF-IDF method, which were 85.9% for the hybrid model and 83.3% for the hybrid model with the attention mechanism applied. Moreover, the TF-IDF method also had the highest performance for all the models tested, except for the BiLSTM method, which achieved its highest score of 80.7% in the GloVe method. The GloVe method was the second in terms of performance, except for the proposed models, which received scores of 80.1% and 81.2% with the attention mechanism applied, which were the inferior scores compared to the scores the scores received from both the TF-IDF and Word2Vec methods.

In addition, the experiments were executed using Word2Vec and GloVe, which were trained on the proposed dataset. In all experiments using the proposed model, the parameters of Word2Vec are min_count = 2, size = 100, window = 5, and sg = 1 using Skip-gram and size = 150, window = 10, min_count = 2, workers = 10, and iter = 50 using CBOW. The models’ performance in terms of accuracy is shown in Table 6.

Table 6. Comparison between accuracy of proposed model using Word2Vec and GloVe.

Word Representation	Word2Vec	GloVe
LSTM (CBOW)	77.2%	-
LSTM (Skip-gram)	78.12%	-
BiLSTM (CBOW)	79.01%	-
BiLSTM (Skip-gram)	80.67%	-
LSTM	-	81.23%
BiLSTM	-	82.45%
Hybrid (CBOW)	84.12%	-
Hybrid (Skip-gram)	86.01%	-
Hybrid with attention (CBOW)	85.11%	-
Hybrid with attention (Skip-gram)	88.73%	-
Hybrid	-	86.12%
Hybrid with attention	-	88.21%

Table 6 indicates the comparison of each model’s performance in terms of accuracy for word representation; the experiment results show that the hybrid with attention Skip-gram (proposed model) offers greater accuracy in comparison with all the other aforementioned models. It was tested with the Word2Vec and GloVe word representation methods, which achieved an 88.73% accuracy rate using hybrid with attention (Skip-gram).

Furthermore, an experiment was conducted to measure and compare the accuracies among four pre-trained state-of-the-art models based on the BERT architecture (MARBERT, AraBert, mBERT, and RoBERTa) and the proposed model presented in this study. All models were tested on the same proposed dataset, and the results of this experiment show us that the proposed model swiftly outperformed all the four pre-trained models in terms

of accuracy since the proposed model achieved an accuracy of 88.73%, while RoBERTa achieved 79.45%, MARBERT attained 76.12%, AraBert reached an accuracy of 73.26%, and lastly, mBERT received an accuracy rate of 69.12%. Therefore, through viewing these statistics, we can confidently say that the proposed model achieved the highest score in terms of accuracy rate as presented in Table 7.

Table 7. Comparison between the accuracy produced using the pre-trained models and the proposed model.

References	Model	Accuracy
[80]	MARBERT	76.12%
[81]	AraBert	73.26%
[82,83]	mBERT	69.12%
[84]	RoBERTa	79.45%
	Proposed Model	88.73%

Overall, the proposed model, which is specially designed to handle sequential data with attention mechanisms, may be very useful for the current job, which is the identification of Arabic dialects on social media. The hybrid model's capacity to distinguish different dialects has been improved by using an attention mechanism. This mechanism enables the model to concentrate on pertinent segments of the input sequence. Although robust and flexible, RoBERTa and GPT-based models are pre-trained on generic text input and might not be as well-suited for this particular task. Furthermore, the model is able to capture various features of the data and take advantage of the advantages of each architecture thanks to the utilization of three parallel branches with distinct designs (LSTM, BiLSTM, and LR). When compared to a single-model method, this combination of models may provide a more thorough comprehension of the input data. The model also concentrates on pertinent segments of the input sequence, the attention self-sequence layers incorporated into each branch. This is especially useful for tasks in which specific words or phrases have a higher weight or relevance. This focused attention process might provide it an advantage when it comes to picking up on minute details in Arabic dialects found in content on social media. Additionally, training procedures tailored to the objective of Arabic dialect detection on social media may improve the performance of the model. The suggested model may outperform pre-trained models like RoBERTa and GPT-based models on this specific job by carefully adjusting hyperparameters, optimizing learning rates, and using suitable training procedures. LSTMs and BiLSTMs are used in capturing long-range dependencies and contextual information, making them ideal for modeling the complex relationships between words in Arabic dialects. Additionally, the bidirectional nature of BiLSTMs allows them to consider both past and future context when making predictions, which is beneficial for dialect classification in which the context of neighboring words is crucial for accurate classification. Furthermore, LSTMs and BiLSTMs can handle variable-length sequences, which is important for Arabic text classification in which sentences can vary greatly in length. In addition, LR is a linear model that works well with high-dimensional data, making it effective for tasks like text classification (Arabic dialect classification) and achieving a high accuracy as an individual classifier

6. Conclusions and Future Work

This study achieves significant improvements over state-of-the-art methods in Arabic dialect detection and classification via a unique hybrid machine and deep learning approach. Our strategy is effective in handling the difficult issues of Arabic dialect recognition, as demonstrated by the hybrid model's higher performance, especially when coupled with attention mechanisms and the TF-IDF word representation method. We have made a contribution by creating a brand new hybrid model for Arabic dialect recognition that performs better than previous models, even those built on the BERT architecture. Furthermore, we have evaluated various word representation techniques, including TF-IDF, Word2Vec, and

GloVe, and have provided insights into how well they work for Arabic dialect identification. The experimental results demonstrate that the proposed model performs more precisely utilizing the modified word representation than the most advanced Arabic dialect detection models currently in use.

Future work should add to the dataset in an effort to improve the accuracy of the model even further. Furthermore, we want to include dialect samples from other Arabic-speaking nations, which should result in a more inclusive and thorough model that can support a wider variety of Arabic dialects. Moreover, to further enhance model performance, we propose examining the usage of extra attention mechanisms or transformer-based designs, investigating how our method might be used for further Arabic natural language processing jobs. Furthermore, using Arabic dialect-specific linguistic traits or contextual data may enhance the model's accuracy and comprehension. Our study, taken as a whole, significantly advances the field of Arabic dialect identification by showcasing the effectiveness of our hybrid model and offering insightful information for further research.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The dataset can be found at <https://www.kaggle.com/datasets/waelshaher/arabic-dialect> (accessed on 4 March 2024).

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Kanan, T.; Sadaqa, O.; Aldajeh, A.; Alshwabka, H.; AL-dolime, W.; AlZu'bi, S.; Elbes, M.; Hawashin, B.; Alia, M.A. A review of natural language processing and machine learning tools used to analyze arabic social media. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; IEEE: Piscataway, NJ, USA; pp. 622–628.
2. Alhejaili, R.; Alhazmi, E.S.; Alsaedi, A.; Yafouz, W.M. Sentiment analysis of the COVID-19 vaccine for Arabic tweets using machine learning. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 3–4 September 2021; pp. 1–5.
3. Alnawas, A.; Arici, N. The corpus based approach to sentiment analysis in modern standard Arabic and Arabic dialects: A literature review. *Politek. Derg.* **2018**, *21*, 461–470. [CrossRef]
4. Al Shamsi, A.A.; Abdallah, S. Text mining techniques for sentiment analysis of Arabic dialects: Literature review. *Adv. Sci. Technol. Eng. Syst. J.* **2021**, *6*, 1012–1023. [CrossRef]
5. Kwaik, K.A.; Saad, M.; Chatzikiyriakidis, S.; Dobnik, S. Shami: A corpus of levantine arabic dialects. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
6. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Process. Manag.* **2020**, *57*, 102121. [CrossRef]
7. Huang, F. Improved arabic dialect classification with social media data. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2118–2126.
8. AlYami, R.; AlZaidy, R. Arabic dialect identification in social media. In Proceedings of the 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 19–21 March 2020; IEEE: Piscataway, NJ, USA; pp. 1–2.
9. Dunn, J. Modeling global syntactic variation in English using dialect classification. *arXiv* **2019**, arXiv:1904.05527.
10. Elfardy, H.; Diab, M. Sentence level dialect identification in Arabic. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Short Papers. Sofia, Bulgaria, 4–9 August 2013; Volume 2, pp. 456–461.
11. Ali, A.; Dehak, N.; Cardinal, P.; Khurana, S.; Yella, S.H.; Glass, J.; Bell, P.; Renals, S. Automatic dialect detection in arabic broadcast speech. *arXiv* **2015**, arXiv:1509.06928.
12. Boujou, E.; Chataoui, H.; Mekki, A.E.; Benjelloun, S.; Chairi, I.; Berrada, I. An open access nlp dataset for arabic dialects: Data collection, labeling, and model construction. *arXiv* **2021**, arXiv:2102.11000.
13. Sobhy, M.; El-Atta AH, A.; El-Sawy, A.A.; Nayel, H. Word Representation Models for Arabic Dialect Identification. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), Abu Dhabi, United Arab Emirates, 8 December 2022; pp. 474–478.
14. El-Haj, M.; Rayson, P.; Aboeazz, M. Arabic dialect identification in the context of bivalency and code-switching. In Proceedings of the 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018; European Language Resources Association: Paris, France, 2018; pp. 3622–3627.

15. Malmasi, S.; Refaee, E.; Dras, M. Arabic dialect identification using a parallel multidialectal corpus. In Proceedings of the International Conference of the Pacific Association for Computational Linguistics, PACLING 2015, Bali, Indonesia, 19–21 May 2015; Springer: Singapore, 2015; pp. 35–53.
16. Butnaru, A.M.; Ionescu, R.T. Unibuckkernel reloaded: First place in arabic dialect identification for the second year in a row. *arXiv* **2018**, arXiv:1805.04876.
17. Johnson, A.; Everson, K.; Ravi, V.; Gladney, A.; Ostendorf, M.; Alwan, A. Automatic dialect density estimation for african american english. *arXiv* **2022**, arXiv:2204.00967.
18. Hassani, H.; Medjedovic, D. Automatic Kurdish dialects identification. *Comput. Sci. Inf. Technol.* **2016**, *6*, 61–78.
19. Nayel, H.; Hassan, A.; Sobhi, M.; El-Sawy, A. Machine learning-based approach for Arabic dialect identification. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 287–290.
20. Mishra, P.; Mujadia, V. Arabic dialect identification for travel and twitter text. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 28 July–2 August 2019; pp. 234–238.
21. Chittaragi, N.B.; Limaye, A.; Chandana, N.T.; Annappa, B.; Koolagudi, S.G. Automatic text-independent Kannada dialect identification system. In *Information Systems Design and Intelligent Applications: Proceedings of Fifth International Conference INDIA 2018 Volume 2*; Springer: Singapore, 2019; pp. 79–87.
22. Doostmohammadi, E.; Nassajian, M. Investigating machine learning methods for language and dialect identification of cuneiform texts. *arXiv* **2020**, arXiv:2009.10794.
23. AlShenaifi, N.; Azmi, A. Arabic dialect identification using machine learning and transformer-based models: Submission to the NADI 2022 Shared Task. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), Abu Dhabi, United Arab Emirates, 8 December 2022; pp. 464–467.
24. Talafha, B.; Farhan, W.; Altakroui, A.; Al-Natsheh, H. Mawdoo3 AI at MADAR shared task: Arabic tweet dialect identification. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 28 July–2 August 2019; pp. 239–243.
25. Mohammed, A.; Jiangbin, Z.; Murtadha, A. A three-stage neural model for Arabic Dialect Identification. *Comput. Speech Lang.* **2023**, *80*, 101488. [[CrossRef](#)]
26. Sundus, K.; Al-Haj, F.; Hammo, B. A deep learning approach for arabic text classification. In Proceedings of the 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; IEEE: Piscataway, NJ, USA; pp. 1–7.
27. Alqurashi, T. Applying a Character-Level Model to a Short Arabic Dialect Sentence: A Saudi Dialect as a Case Study. *Appl. Sci.* **2022**, *12*, 12435. [[CrossRef](#)]
28. Abdelazim, M.; Hussein, W.; Badr, N. Automatic Dialect identification of Spoken Arabic Speech using Deep Neural Networks. *Int. J. Intell. Comput. Inf. Sci.* **2022**, *22*, 25–34. [[CrossRef](#)]
29. Fares, Y.; El-Zanaty, Z.; Abdel-Salam, K.; Ezzeldin, M.; Mohamed, A.; El-Awaad, K.; Torki, M. Arabic dialect identification with deep learning and hybrid frequency based features. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 28 July–2 August 2019; pp. 224–228.
30. Elaraby, M.; Abdul-Mageed, M. Deep models for arabic dialect identification on benchmarked data. In Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018), Santa Fe, NM, USA, 20 August 2018; pp. 263–274.
31. Mekki, A.E.; Mahdaouy, A.E.; Essefar, K.; Mamoun, N.E.; Berrada, I.; Khoumsi, A. BERT-based Multi-Task Model for Country and Province Level Modern Standard Arabic and Dialectal Arabic Identification. *arXiv* **2021**, arXiv:2106.12495.
32. Wang, J.H.; Liu, T.W.; Luo, X.; Wang, L. An LSTM approach to short text sentiment classification with word embeddings. In Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018), Hsinchu, Taiwan, 4–5 October 2018; pp. 214–223.
33. Nowak, J.; Taspinar, A.; Scherer, R. LSTM recurrent neural networks for short text and sentiment classification. In *Proceedings of the Artificial Intelligence and Soft Computing: 16th International Conference, ICAISC 2017, Zakopane, Poland, 11–15 June 2017*; Proceedings, Part II 16; Springer International Publishing: Cham, Switzerland, 2017; pp. 553–562.
34. Elaraby, M.; Zahran, A. A Character Level Convolutional BiLSTM for Arabic Dialect Identification. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 28 July–2 August 2019; pp. 274–278.
35. Alhazzani, N.Z.; Al-Turaiki, I.M.; Alkhodair, S.A. Text Classification of Patient Experience Comments in Saudi Dialect Using Deep Learning Techniques. *Appl. Sci.* **2023**, *13*, 10305. [[CrossRef](#)]
36. De Francony, G.; Guichard, V.; Joshi, P.; Afli, H.; Bouchekif, A. Hierarchical deep learning for Arabic dialect identification. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 28 July–2 August 2019; pp. 249–253.
37. Lulu, L.; Elnagar, A. Automatic Arabic dialect classification using deep learning models. *Procedia Comput. Sci.* **2018**, *142*, 262–269. [[CrossRef](#)]
38. Althobaiti, M.J. Country-level Arabic dialect identification using small datasets with integrated machine learning techniques and deep learning models. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 265–270.
39. Mansour, M.; Tohamy, M.; Ezzat, Z.; Torki, M. Arabic dialect identification using BERT fine-tuning. In Proceedings of the Fifth Arabic Natural Language Processing Workshop, Barcelona, Spain, 12 December 2020; pp. 308–312.
40. Yahya, A.E.; Gharbi, A.; Yafooz, W.M.; Al-Dhaqm, A. A Novel Hybrid Deep Learning Model for Detecting and Classifying Non-Functional Requirements of Mobile Apps Issues. *Electronics* **2023**, *12*, 1258. [[CrossRef](#)]

41. Abdul-Mageed, M.; Zhang, C.; Elmadany, A.; Bouamor, H.; Habash, N. NADI 2022: The Third Nuanced Arabic Dialect Identification Shared Task. *arXiv* **2022**, arXiv:2210.09582.
42. Abdelali, A.; Mubarak, H.; Samih, Y.; Hassan, S.; Darwish, K. Arabic dialect identification in the wild. *arXiv* **2020**, arXiv:2005.06557.
43. Alghamdi, A.; Alshutayri, A.; Alharbi, B. Deep Bidirectional Transformers for Arabic Dialect Identification. In Proceedings of the 6th International Conference on Future Networks & Distributed Systems, Tashkent, Uzbekistan, 15 December 2022; pp. 265–272.
44. Attieh, J.; Hassan, F. Arabic Dialect Identification and Sentiment Classification using Transformer-based Models. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), Abu Dhabi, United Arab Emirates, 8 December 2022; pp. 485–490.
45. Fsih, E.; Kchaou, S.; Boujelbane, R.; Belguith, L.H. Benchmarking transfer learning approaches for sentiment analysis of Arabic dialect. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), Abu Dhabi, United Arab Emirates, 8 December 2022; pp. 431–435.
46. Messaoudi, A.; Fourati, C.; Haddad, H.; BenHajhmida, M. iCompass Working Notes for the Nuanced Arabic Dialect Identification Shared task. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), Abu Dhabi, United Arab Emirates, 8 December 2022; pp. 415–419.
47. Talafha, B.; Ali, M.; Za’ter, M.E.; Seelawi, H.; Tuffaha, I.; Samir, M.; Farhan, W.; Al-Natsheh, H.T. Multi-dialect arabic bert for country-level dialect identification. *arXiv* **2020**, arXiv:2007.05612.
48. Bayrak, G.; Issifu, A.M. Domain-Adapted BERT-based Models for Nuanced Arabic Dialect Identification and Tweet Sentiment Analysis. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), Abu Dhabi, United Arab Emirates, 8 December 2022; pp. 425–430.
49. Beltagy, A.; Wael, A.; ElSherief, O. Arabic dialect identification using bert-based domain adaptation. *arXiv* **2020**, arXiv:2011.06977.
50. Bouamor, H.; Habash, N.; Salameh, M.; Zaghrouani, W.; Rambow, O.; Abdulrahim, D.; Obeid, O.; Khalifa, S.; Eryani, F.; Erdmann, A.; et al. The madar arabic dialect corpus and lexicon. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
51. Abdul-Mageed, M.; Zhang, C.; Bouamor, H.; Habash, N. NADI 2020: The first nuanced Arabic dialect identification shared task. *arXiv* **2020**, arXiv:2010.11334.
52. Abdul-Mageed, M.; Zhang, C.; Elmadany, A.; Bouamor, H.; Habash, N. NADI 2021: The second nuanced Arabic dialect identification shared task. *arXiv* **2021**, arXiv:2103.08466.
53. Abdul-Mageed, M.; Elmadany, A.; Zhang, C.; Nagoudi, E.M.B.; Bouamor, H.; Habash, N. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. *arXiv* **2023**, arXiv:2310.16117.
54. Abdelali, A.; Mubarak, H.; Samih, Y.; Hassan, S.; Darwish, K. QADI: Arabic dialect identification in the wild. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 1–10.
55. Bouamor, H.; Habash, N.; Oflazer, K. A Multidialectal Parallel Corpus of Arabic. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14), Reykjavik, Iceland, 26–31 May 2014; pp. 1240–1245.
56. Alsarsour, I.; Mohamed, E.; Suwaileh, R.; Elsayed, T. Dart: A large dataset of dialectal arabic tweets. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
57. Althobaiti, M.J. Automatic Arabic dialect identification systems for written texts: A survey. *arXiv* **2020**, arXiv:2009.12622.
58. Etman, A.; Beex, A.L. Language and dialect identification: A survey. In Proceedings of the 2015 SAI intelligent systems conference (IntelliSys), London, UK, 10–11 November 2015; IEEE: Piscataway, NJ, USA; pp. 220–231.
59. Harrat, S.; Meftouh, K.; Smaili, K. Maghrebi Arabic dialect processing: An overview. *J. Int. Sci. Gen. Appl.* **2018**, *1*, 38.
60. Harrat, S.; Meftouh, K.; Smaili, K. Machine translation for Arabic dialects (survey). *Inf. Process. Manag.* **2019**, *56*, 262–273. [[CrossRef](#)]
61. Elnagar, A.; Yagi, S.M.; Nassif, A.B.; Shahin, I.; Salloum, S.A. Systematic literature review of dialectal Arabic: Identification and detection. *IEEE Access* **2021**, *9*, 31010–31042. [[CrossRef](#)]
62. Issa, E.; AlShakhori, M.; Al-Bahrani, R.; Hahn-Powell, G. Country-level Arabic dialect identification using RNNs with and without linguistic features. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 276–281.
63. Baimukan, N.; Bouamor, H.; Habash, N. Hierarchical aggregation of dialectal data for Arabic dialect identification. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; pp. 4586–4596.
64. Obeid, O.; Inoue, G.; Habash, N. Camelira: An Arabic multi-dialect morphological disambiguator. *arXiv* **2022**, arXiv:2211.16807.
65. Tzudir, M.; Baghel, S.; Sarmah, P.; Prasanna, S.R.M. Analyzing RMFCC Feature for Dialect Identification in Ao, an Under-Resourced Language. In Proceedings of the 2022 National Conference on Communications (NCC), Mumbai, India, 24–27 May 2022; IEEE: Piscataway, NJ, USA; pp. 308–313.
66. Shon, S.; Ali, A.; Samih, Y.; Mubarak, H.; Glass, J. ADI17: A fine-grained Arabic dialect identification dataset. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA; pp. 8244–8248.
67. Rong, X. word2vec parameter learning explained. *arXiv* **2014**, arXiv:1411.2738.
68. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

69. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
70. Zhang, S.; Zheng, D.; Hu, X.; Yang, M. Bidirectional long short-term memory networks for relation classification. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China, 30 October–1 November 2015; pp. 73–78.
71. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [[CrossRef](#)]
72. Jang, B.; Kim, M.; Harerimana, G.; Kang, S.U.; Kim, J.W. Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism. *Appl. Sci.* **2020**, *10*, 5841. [[CrossRef](#)]
73. Bae, K.; Ryu, H.; Shin, H. Does Adam optimizer keep close to the optimal point? *arXiv* **2019**, arXiv:1911.00289.
74. Şen, S.Y.; Özkurt, N. Convolutional neural network hyperparameter tuning with adam optimizer for ECG classification. In Proceedings of the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 15–17 October 2020; IEEE: Piscataway, NJ, USA; pp. 1–6.
75. Aghaebrahimian, A.; Cieliebak, M. Hyperparameter tuning for deep learning in natural language processing. In Proceedings of the 4th Swiss Text Analytics Conference (Swisstext 2019), Winterthur, Switzerland, 18–19 June 2019.
76. Yafooz, W.; Alsaeedi, A. Leveraging User-Generated Comments and Fused BiLSTM Models to Detect and Predict Issues with Mobile Apps. *Comput. Mater. Contin.* **2024**, *79*, 735–759. [[CrossRef](#)]
77. Sari, W.K.; Rini, D.P.; Malik, R.F. Text Classification Using Long Short-Term Memory with GloVe. *J. Ilm. Tek. Elektro Komput. Dan Inform. (JITEKI)* **2019**, *5*, 85–100. [[CrossRef](#)]
78. Ruby, U.; Yendapalli, V. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng.* **2020**, *9*, 5393–5397.
79. Zhang, C.; Woodland, P.C. Parameterised sigmoid and ReLU hidden activation functions for DNN acoustic modelling. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.
80. Abdul-Mageed, M.; Elmadany, A.; Nagoudi, E.M.B. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. *arXiv* **2020**, arXiv:2101.01785.
81. Antoun, W.; Baly, F.; Hajj, H. Arabert: Transformer-based model for arabic language understanding. *arXiv* **2020**, arXiv:2003.00104.
82. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019.
83. Pires, T.; Schlinger, E.; Garrette, D. How multilingual is multilingual BERT? *arXiv* **2019**, arXiv:1906.01502.
84. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.