Classifications of Arabic Customer Reviews Using Stemming and Deep Learning

Hawraa Fadhil Khelil[1], Mohammed Fadhil Ibrahim[2], Hafsa Ataallah Hussein[3]
[1,2,3] Technical College of Management, Middle Technical University (MTU), Baghdad, Iraq

Article Info

Article history:

Received Feb 18, 2024
Revised Jun 25, 2024
Accepted Jul 18, 2024

Keyword:

Customer Review
Arabic Text Classification
Text Stemming
Deep Learning
Text Feature Extraction

ABSTRACT

With the emergence of AI text-based tools and applications, the need to present and investigate text-processing tools has also been raised. NLP tools and techniques have developed rapidly for some languages, such as English. However, other languages, like Arabic, still need to present more methods and techniques to present more explanations. In this study, we present a model to classify customer reviews which are written in Arabic. The HARD dataset is used to be adopted as the dataset. Three Deep Learning classifiers are adopted (CNN, LSTM, RNN). In addition to that, three stemmers are used as text processing techniques (Khoja, Snowball, Tashaphyne). Furthermore, another three feature extraction methods were utilized (TF-IDF, N-gram, BoW). The results of the model presented several explanations. The best performance resulted from using (CNN+ Snowball+ N-Gram) with an accuracy of (%93.5). The results of the model stated that some classifiers are sensitive toward using different stemmers, also some accuracy performance can be affected if there are different feature extraction methods used. Either stemming of feature extraction has an impact on the accuracy performance. The model also proved that the dialectal language could cause some limitations since different dialects can give conflict meaning across different regions or countries. The outcomes of the study open the gate towards investigating other tools and methods to enrich Arabic natural language processing and contribute to developing new applications that support Arabic content.

Corresponding Author:

Mohammed Fadhil Ibrahim,
Technical College of Management,
Middle Technical University (MTU),
Safi Al Din Al Hilli Street, Baghdad, Iraq
Email: mfi@mtu.edu.iq

1. INTRODUCTION

In computer science, the discipline of natural language processing (NLP) seeks to enable communication between humans and machines (computers that can comprehend machine language, such as English, Arabic, Chinese, etc.). NLP is vital because it dramatically impacts our daily lives [1]. Many business operations can be made simpler and more automated with the use of (NLP), particularly those that involve a lot of unstructured text, including emails, surveys, social media chats, etc. Businesses may more effectively evaluate their data with NLP to support wise decision-making. It is a fundamental subject in (AI), and its applications, such as popular machine translation, speech recognition, public opinion analysis, text classification, and so on [2]-[4]. In NLP research, the typology of the text plays a significant role. Text classification is one of the crucial areas of NLP. The classification problem has been extensively studied in machine translation, database, and information retrieval, with applications in many fields, such as targeted marketing, medical diagnostics, newsgroup filtering, document organization, and sentiment analysis [5], [6]. It is known that it is challenging to address the best text classifier [7]. Text classification still lacks a benchmark method, primarily when texts are written in Arabic.

One of the world's oldest and richest languages, Arabic is also one of the most commonly spoken languages worldwide and is a cornerstone of humankind's cultural variety. It is the fifth most widespread language on the Internet and the fourth most popular language in the world. Arabic is spoken by more than 6.6% of the global population [8]. The nature of Arabic as a macro-morphological language with many varieties makes it one of the most complicated scripts [9]. IOS Registration Authority has classified it as a language with 30 varieties, including Modern official standard Arabic. For example, the word (عقد) might mean (contract). Still, Arabic has a variety of diacritics annotations placed with word letters that control the meaning, so the general figure of the word doesn't give the exact meaning. Many times, the meaning depends on the context; if the diacritics were omitted, different connotations in meaning. Suppose the words (عَقَدَ – عَقَّدَ - عُقَّدْ عُقْد– عِقد - عَقْد), which mean (Necklace –Decade –Contract – Held – Complicated –Knots) respectively, so that the past word seems similar in terms of shape but gives different meanings when considering diacritics. Text classification can be divided into two categories, the first of which is based on traditional machine learning methods, such as support vector machine (SVM) [10], [11], Naive Bayes [12], decision tree [13], K-Nearst-Neighbours (KNN) [14], maximum entropy and so on. The main problems of this search method are that the processed text generally has the disadvantages of high dimension and sparse vector, which facilitates the production of a gradient burst or gradient disappearance in the follow-up search; simple machine learning methods take a long time and are generally difficult to operate, the second category is the use of deep learning methods where convolutional neural network and repetitive neural network are typically used.

Initially, applied DL methods were gradually spread in the text field with further research development. So, there has been a considerable amount of research considering NLP methods and techniques for vital language. Still, the number of research dealing with the Arabic language is below the level of ambition if compared to the language's importance and widely spread. To cope with this gap, it is essential to provide more sophisticated methods and techniques to process Arabic texts and analyze the content of reviews, especially with the increasing use of social media and internet-based applications. Thus, this study presents a model of Arabic review classification using stemming techniques along with feature extraction, and for the classification, this study employs three DL techniques. The rest of this research is organized as follows: section two is assigned to view the most related works, section three is depicted to describe the research methodology, and section four is dedicated to describing the classification process. Section five describes the implementation and results discussion, and finally, section six shows the study's conclusions.

2. RELATED WORKS

This part aims to discuss some important previous studies in the processing and classifying of texts written in Arabic, which is the basic starting point for this work, and to understand the research topic, its requirements, and goals; in this part, we discuss some of the research related to the topic of processing and classification of Arabic texts by the methods used in word processing and the algorithms that were used. In [15], the authors proposed the "An Easier Data Augmentation AEDA" technique for text classification to help improve performance in text classification with RNN CNN algorithms. AEDA only includes random insertion of punctuation marks in the original text. The study showed this by using AEDA-enhanced data for training, where models have shown superior performance compared to other models. [16] Almuzaini and others used seven DL algorithms to classify Arabic documents: CNN, CNN-LSTM (LSTM = Long-Range Short-Range Memory), CNN-GRU (Gated Recurrent Unit), BiLSTM, and ATT-LSTM. For word representation, they applied word embedding technology (Word2Vec). Algorithms were tested on two large sets of data. The best performance was the F-score, achieved using the ATT-GRU model with the trunk-based algorithm. Zhang and others suggested expanding the non-negative matrix analysis feature to overcome the features' scattering in short texts [17]. Given two sets: a word set and a short text set. K groups are created from each set. The relationship between types of short texts and words is described by the author's definition of the matrix R. Additionally, two matrices were employed to illustrate the link between words and texts inside a type. In addition, the indicator was compiled using two extra matrices. Consequently, cooperatively grouped texts and words are produced using the short text expansion function. Three datasets were used to evaluate their method, including Twitter sports, which had six categories (such as baseball, basketball, and football). The approach beat Word2vec's accuracy by 10.89% and CNN's character-level usage (CNN) by 32%. A study presented in [18] has implemented two deep learning models, LSTM and CNN, as well as three traditional techniques: Naïve Bayes, KNN, and decision trees to analyze emotions and compare experimental results. It also offers a built-in model from CNN and RNN structure, where this model collects local features through CNN as an RNN input for Arabic sentiment analysis of short texts. Appropriate data reparation has been made for each data set used. Experiments were conducted for each dataset against the traditional machine learning classification, KNN, NB, tree resolution, and regular deep learning models; CNN and LSTM have resulted in good performance.

In [19], Bedir and Ibrahim used the CNN and RNN DL algorithms to categorize Arabic tweets. With Twitter API, the writers gathered 160,870 Arabic tweets. Basketball, football, criminal accidents, traffic accidents, vocalists, beauty and fashion, technology, and economy are the eight areas into which the data collection is separated. The distribution of tweets for the remaining categories was more balanced, with the exception of 8600 traffic accidents. Only 110,000 tweets, or 90% of the data set, were left for testing by the authors after they trained and validated DL models on 144,000 tweets. The DL models performed extremely similarly. In [20], Guangquan and others have used two methods along with CNN: a duplicate unit with two-way gates and an attention network unit. In particular, the attention unit allows the method to learn and represent special features in local accreditation of training texts, and CNN can learn global representation when participating. Experiments with 16 subsets of Amazon audit data show that the method outperforms many baselines and also proves the effectiveness of multi-pedigree co-learning tasks. Walid Cherif and others have offered an alternative way to process text classification [21]. First, it reduces the original feature set using a newly proposed scale. Second, the text is automatically classified without necessarily addressing all its features. Also, some standard pre-treatments, such as eradication, can be abandoned. Empirical results have shown that this new text classification method is superior to modern methods. As a result, the metrics obtained in the 20 news data sets, BBC News, Reuters, and AG, were in suitable proportions, while standard methods gave much lower scores. In [22], an efficient technique for feature selection based on extracting association rules for text document classification was proposed by Saeed & Al Aghbari. The linkages and correlations between the pertinent terms within and between text documents within a category are found using the rules of association that are extracted. As a result, a limited number of conflicting characteristics that are better at text classification serve as the representation for each category of documents. Experiments have demonstrated that in terms of classification performance and efficiency, ARTC performs better than other pertinent technologies.

While Alzanin and others [23] implemented three different methods: SVM, Gaussian Naïve Bayes (GNB), and Random Forest (RF), each method has set superior parameters. They collected a dataset of about 35,600 Arabic languages and manually commented on their tweets for experiments. Statistically, radio frequencies and SVM with radial foundation function nucleus (RBF) performed equally well when used with stemming and TF-IDF. GNB, with word embedding performance, was disappointingly low. In order to reduce the workload for humans, Sunil & Dong presented a technique for classifying medical texts utilizing two DL structures [24]. The first method uses four channels to implement the hybrid DL model for long-term memory (QC-LSTM) quadrilateral; the second method involves developing and successfully implementing the DL model of the biGRU with multi-head interest. Two sets of medical text data were used to validate the suggested methodology, and a thorough analysis was carried out. The deep QC-LSTM technology that was suggested produced the best rating accuracy results.

From the above studies, we can develop some implications related to Arabic text classification that affect the results, whether regular ML or DL. Whether long or short, text size is essential for obtaining good performance. Best text sentiment analysis techniques can understand, analyze, and classify texts accurately. The accuracy gained is high when the processing process is developed and relies on using more sophisticated data-cleaning methods or extracting features. Texts written in Arabic require more text-processing techniques than in other languages because a language’s morphological composition varies, in addition to the dialects, which also vary from one region to another in Middle Eastern communities. Hence, many researchers reported that traditional processing methods may suffer from low accuracy in performance when used with Arabic scripts. In this research, more experiments will be implemented to investigate using different stemming tools along with DL algorithms to support past research and come out with new explanations.

3. METHODOLOGY

Generally speaking, the majority of NLP investigations follow a few standard procedures that illustrate the entire research procedure. Our study’s approach is shown in Figure 1. Data collection for the study is the first step, followed by pre-processing tools and methods, and model evaluation is the last.

3.1. Data Collection.

Arabic-language hotel reviews can be found in the dataset utilized for this work, the Hotel Arabic Reviews Dataset (HARD) [25]. A number of studies and research projects involving Arabic language and NLP have made use of the HARD dataset [26 - 28]. Between June and July of 2016, this dataset was gathered from the Booking.com website. About 93,700 evaluations of positive and negative classes are included in the first balanced HARD dataset (

). In terms of Arabic, Modern Standard Arabic (MSA), the formal language, and Dialectal Arabic (DA), which is colloquial Arabic, customer reviews were inconsistent. Every region of the Arabic countries has a different definition for DA, and each country's official Arabic language has multiple variants according to its location [29].

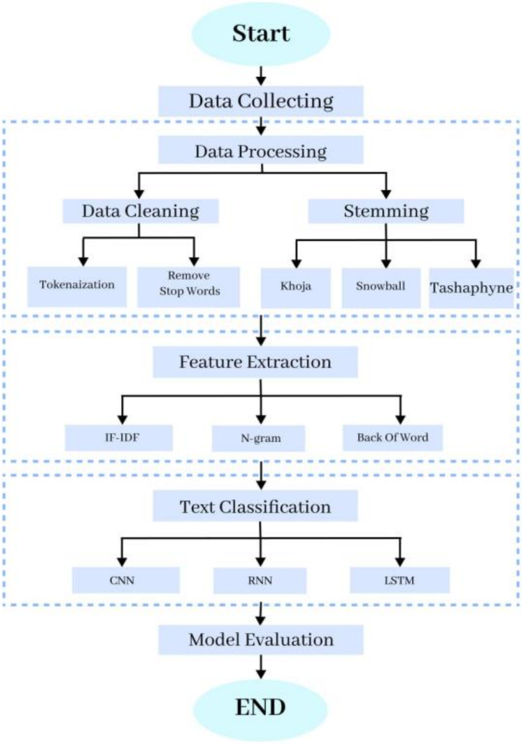

Figure 1. Research Framework

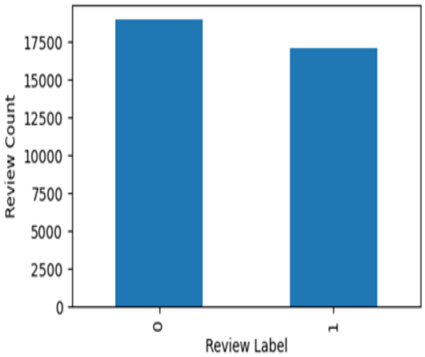

Figure 2. Dataset Description

Through the representation of the dataset, it was demonstrated that there is a significant degree of variation in the text length of the customer reviews (Figure 3). Thus, while some people record a few words, others register lengthy comments. Short texts generally suffer from NLP's drawbacks because there is less information contained in them. As a result, the data set underwent an early stage of removal of the lengthy text (more than 800 characters) and the succinct remarks (less than 100 characters). Following this stage, the dataset grows to 36098, and reviews are categorized as Positive (17106) and Negative (18992).

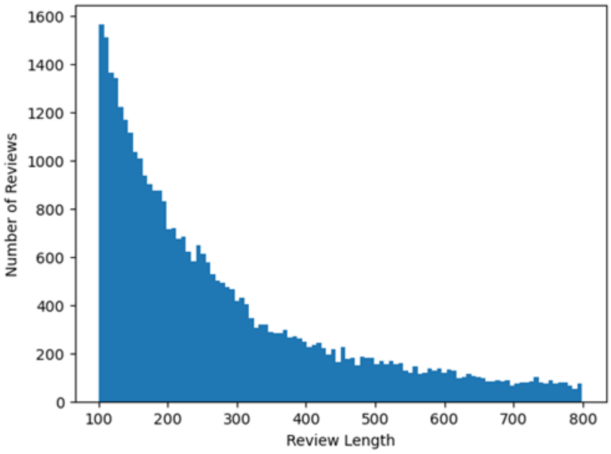

Figure 3. Customer Reviews Distribution Based on Text Length

3.2. Data Pre-processing

Setting up the data for the task of classification is one of the primary steps in any NLP-related work. A few steps are involved in this process to prepare the data for additional analysis. Performing similar methods for any text has some restrictions since the quality of the information being processed primarily determines its output efficiency. Arabic is one of the most complicated scripts and is quite sensitive to operations, including encoding and cleaning [30]. Dealing with the diversity of dialects, their highly derivative character, and the uncertainty brought on by diacritical marks all add to this complexity. For corporate applications to yield robust, accurate, and dependable results, data pre-processing is a necessary step in almost all data analysis, data science, and artificial intelligence development processes. Real-world data is chaotic and is produced, handled, and retained by a variety of people, organizations, and software programs. Consequently, the dataset can have different names for the same entity, contain errors in human entry, have duplicate data, or be missing entire fields. In many cases, humans are able to recognize and fix these issues in the data that they utilize for their work. Yet, automatic pre-processing of the data is required for deep learning algorithms or machine learning training [3]. Restructuring unprocessed information into a model appropriate for a given method is made possible by features engineering methods that include processing the data, modification, reduction, selecting features, and scaling. This can drastically cut down on the amount of time and processing power needed to train a new AI or machine learning algorithm or compare a result to it. Pre-processing techniques used in this study for Arabic texts included the following:

A. Tokenization

The first among the initial steps in every NLP processing chain is coding [31]. It breaks up unprocessed text into tiny sentences or word pieces known as tokens. Generally speaking, characters like “dots, exclamation marks, and newline characters” and word encoding are performed using the “space” character. The particular NLP task at hand determines which coding technique is best [32]. This method breaks down textual content into its constituent words.

B. Removing Stopwords

Prepositions, sign names, hyphenated nouns, and interrogative tools are examples of words that are categorized as stop words since they have no meaningful relationship to the context in which they are employed [33]. Stopwords are terms that are commonly found in the majority of documents within a particular group. It might not appear like these frequently used terms matter when deciding whether postings meet the user’s needs. Put differently, as these terms appear in both favourable and bad reviews, they cannot help distinguish between the two [34]. Therefore, aside from dimensionality reduction -which is crucial for the majority of machine learning tasks- these terms have little bearing on the classification problem and can be eliminated without impairing the performance of the classifier Table .

Table 1. Sample Stop words in the Arabic language

Arabic Word	English Meaning	English Pronunciation	Category
في	In	Fey	**Prepositions**
على	on	Alaa	
الى	to	Ela	
أنا	I	Ana	**Pronouns**
نحن	we	Nahno	
تحت	Below	Taht	**Adverbs**
فوق	Above	Fawk	
الان	Now	Al’an	
منذ	Since	Monzo	
لماذا	What	Lemaza	**Question**
متى	When	Mata	
اذا	If	Eza	**Conjunctions**
ثم	Then	Soma	
عدا	except	Ada	

C. Stemming

The stem word is an important feature supported by today’s indexing and search systems [35]. Indexing and searching are, in turn, part of text extraction applications, NLP systems, and Information Retrieval Systems. Combining related words into a single word is the basic notion behind stemming. For instance, the term “Histori” would take the place of both “Historical” and “History”. The important thing to remember is that every time a word comes in this form, there is a chance that it will be understood similarly. Although the word “Histori” in the previous example has no meaning in the English language, it can still be used to classify a specific text.

For this reason, stemming can be helpful in solving classification problems but may not produce the necessary results for other NLP applications. When working with an alphabet as complicated as Arabic, the use of the stemming approach effectively produces terms that are identical to the same root [36]. The fundamental objective of a root-based root is to determine a word's main structure through morphological study. The scenario is exemplified in (Table 1). Here, it can be seen that the term in Arabic can have many forms of writing, which in most cases is noticeably different from the original form, and even can have different shapes in terms of characters used. If it is compared to English, then we can see the complexity of Arabic scripts surpasses English and other languages.

Table 1. Example of Arabic Terms Complexity

Term	Possible Forms	English meaning
	لعبة	Game
	ملعب	Stadium
The Arabic word "يلعب" means (Play) in English	لاعب	Player
	لاعبان	Two Players (Males)
	لاعبتان	Two Players (Females)
	لاعبات	Players (Females)
	لاعبين	Players (Males)

This example demonstrates the intricacy and diversity of Arabic letters. Such words can be difficult to decode and analyze, and because Arabic does not adhere to a set orthographic style based on letters, businesses may have a high error rate. Alternatively, a word with a similar meaning to the original one may be clearly different. Thus, the previous example and all of its related terms can be reconstructed from stems to a single root, such as "لعب", which means "play" [37].

In this work, three distinct stemmers are used, and their efficiency is assessed correspondingly. First, Khoja Stemmer is used. Khoja's approach is one of the most popular morphological extraction methods [31], [38]. Khoja Stemmer eliminates a word's longest prefix and suffix. The word's root is then found by matching the remainder of the words to nouns and linguistic patterns. The method evaluates the remaining part of the term using its verbal and conceptual features to identify the root after removing the largest prefix and suffix from the word [39], [40]. Second, the Porter Stemmer algorithm has been improved upon by Snowball Stemmer, which uses the extraction algorithm, also known as the porter2 derivation algorithm, in order to address a number of its shortcomings.

To conclude, Tashaphyne Stemmer is a syllable and mild Arabic derivation [41]. Its main goal is to promote light derivation by removing prefixes and suffixes and offering every possible division. It uses a modified finite state automaton to produce all of the partitions[42], [43].

It provides concurrent root retrieval and extraction, as well as ESRI, Asim, and Frasa stemmers, in contrast to Khoja. Tashaphyne can handle extra features and create custom derivatives without necessitating modifications to the code because it permits utilizing a list of customized suffixes and prefixes as opposed to having them by default [43], [44]. To clarify the differences between the three stemmers,

Table 2 presents some examples of how stemmers can manipulate Arabic texts. We notice the instance of the word ("الهدوء"), which means "Quietness," can be processed differently across different stemmers. Despite each stemmer having its processing outputs, they share some manipulation features. For example, all stemmers remove the Arabic prefixes, such as the prefix ("الـ"), which is equivalent to the prefix ("the") in English.

Table 2. Arabic Stemming Example

Original Text	Khoja	Snowball	Tashaphyne	English Meaning
الهدوء	هدء	هدوء	هدء	The quietness
المطعم	طعم	مطعم	طعم	The restaurant
التكييف جيد	كيف جيد	تكييف جيد	كوف جيد	Air conditioning is good
جيد لابأس	جيد بأس	جيد لاباس	جيد بءس	Good it's okay

3.3. Feature extraction

Feature extraction is part of the dimensionality reduction process, in which an initial raw data set is divided and reduced into more manageable groups, making the process more straightforward [45]. These huge data sets' numerous variables are their most crucial feature. Processing these variables takes a lot of computing resources. Consequently, feature extraction efficiently reduces the amount of data by choosing and combining variables into features, helping to extract the best features from such large data sets. These characteristics are simple to handle while accurately and creatively describing the real data set. We shall use multiple feature extraction methods in this aspect [46].

A. TF-IDF:

During the pre-processing phase, text features such as keyword information are extracted from the text using TF-IDF, stop word removal, and word segmentation [47], [48]. A statistical technique for determining a word's meaning in a text is called TF-IDF. The significance of a word rises in direct proportion to how frequently it appears in the document, but it falls in proportion to how frequently it appears in the corpus. Some discernible words can be identified more accurately by TF-IDF. In some works, these words are used more frequently than in others. We apply the following formula to determine the TF-IDF value of each word in the lexicon of test sample K, allowing us to determine the frequencies of all the words:

$$TF - IDF = TF * IDF \qquad \ldots (1)$$

Where *TF* represents the number of occurrences for any given term in a specific text divided by the total number of terms in a given text, *IDF* represents the logarithmic value of the number of texts in the dataset divided by the number of texts in the dataset.

The unique ratings for every word in the text are extracted using this formula. Higher-valued words are more selective and best differentiated between text groups. By keeping the terms that are most helpful in categorization, these words reduce the quantity of text and the number of attributes that must be computed. TF-IDF arranges words in distinct texts based on their significance, sorting them from largest to lowest. Greater-meaning words have more discriminative power for each text because words with greater weights in one text have less weight in other texts. To precisely extract the most essential text features and eliminate unnecessary features, we employ TF-IDF [49].

B. N-grams

Contiguous groups of "n" items -usually words or characters- taken from a text corpus are known as "N-grams." These models, which provide an easy way to capture the statistical features of the language, have been essential to both early and modern Natural Language Processing. Because N-gram models capitalize on local dependencies within a text, they are particularly well-suited for jobs involving short-range contextual information. When attempting to estimate a word's likelihood based on its previous context, n-gram models are frequently used in language modeling tasks. Every word in a unigram model is handled separately, but bigram models take into account word pairings. Higher-order models and trigrams can capture longer dependencies. N-gram models are helpful when more complicated models are computationally expensive, even though they provide a foundation for language modelling despite their simplicity [50], [51].

C. Bag-Of-Words (BoW)

When extracting features from the text for modeling purposes, such as machine learning techniques, the result is a bag-of-words model or BoW for short. The method is incredibly versatile and easy to use, and it may be applied in many ways to extract features from the text. A BoW is a text representation that lists words in a corpus according to their occurrences. It has two components: a list of well-known words in one's vocabulary and a gauge of the quantity of known words. It is referred to as a "bag" of words since the document discards any information regarding the word order or structure. The model does not care where recognized words appear in the document; it only cares whether they do [52], [53].

The BoW technique is a widely used feature extraction procedure for sentences and texts. When using this method, each word count is considered as a feature, and the word histogram within the text is examined. It makes sense that texts with comparable content would be similar; furthermore, it can infer some information about the document's meaning just from its content. It is possible to choose how basic or sophisticated the BoW is. The intricacy lies in the decisions on how to create the vocabulary of recognized words (or tokens) and how to assign a score for their appearance [54], [55].

3.4. Texts classification

One of the key components of NLP is the classification of texts [56]. The best text classifier cannot be defined, as is well known. For instance, there is broad agreement on a standard approach for creating models, neural networks in particular, and other recognized methodologies in fields like computer vision [57]. Besides, this common approach continues to be lacking in many aspects of text classification. Since Arabic information on the Internet is growing at an ongoing rate, one of the key themes in large-scale Arabic text mining is the classification of Arabic texts. It is one of the most significant study areas, where excellent data is taken from texts, and the themes to which those texts belong are categorized, particularly when these texts [58]. The vast number of Arabic texts available on the Internet leads to scholarly problems that have recently been addressed. Researchers are trying to make use of this data by classifying the texts using methods such as data mining. The objective of this study was to apply the beneficial effects of algorithms that have been effective across various Arabic language domains. In this study, we will be covering certain algorithms, namely CNN, RNN, and LSTM.

A. Convolution Neural Network (CNN)

Although early 2D CNNs were applied extensively in computer vision, however, text classification tasks are a relatively new application for them, and they have shown superior performance than sequence-based methods [59]. Using a convolutional layer and a subsampling layer (also known as the maximum pooling layer), the CNN creates a feature map through a series of convolutions and pooling [60]. A sliding convolution window with changing kernels is used by the convolution layer of the 1D CNN to perform a 1D cross-correlation operation across the text being input from left to right [61]. It uses a max-over-time pooling layer, which lowers the amount of features required for text encoding by using a 1D global maximum pooling layer.

B. Recurrent Neural Network (RNN)

RNNs are widely used in scenarios involving sequential data. This is because the model uses layers, which offer short-term memory. It can predict the subsequent data more accurately thanks to this memory [62]. The period of the past data's retention is determined by its associated weight is a dynamic process [63]. Consequently, speech marking, sign sequence analysis, emotion analysis, and other applications need RNN. By far, the most significant advantage of an RNN is its capacity to communicate previous knowledge to the latest output, i.e., to connect the current production of the series with the last one. Text analysis is improved by the bidirectional repetitive neural network's ability to correlate context semantics in word processing accurately.

C. Long Short-Term Memory (LSTM)

An LSTM network is an instance of a recurrent neural network (RNN), which is a collective term for a group of neural networks with sequential data processing capabilities [64]. Three “gate” components make up the distinctive network structure known as LSTM [61]. An LSTM has three gates termed input gates; the output and gate gates are not present. Despite entering the LSTM network, the data might be chosen based on predetermined criteria [18]. The forget gate causes the non-matching data to be erased, leaving just the data that matches the algorithm.

4. IMPLEMENTATION AND RESULTS DISCUSSION

As stated earlier, the model's efficiency was assessed using three classification algorithms: CNN, RNN, and LSTM. The results of using the classifiers on three stemming approaches (Khoja, Snowball, and Tashaphyne) and three methods for extracting features (TF-IDF, N-grams, BoW) have been used to evaluate the classification performance. There will be variations amongst stemmers in terms of rooted each word because every stemming technique has its protocols for stemming the Arabic language. Different classifiers should produce different results based on that. Figure 4 describes the network architecture for all of the classifiers used.

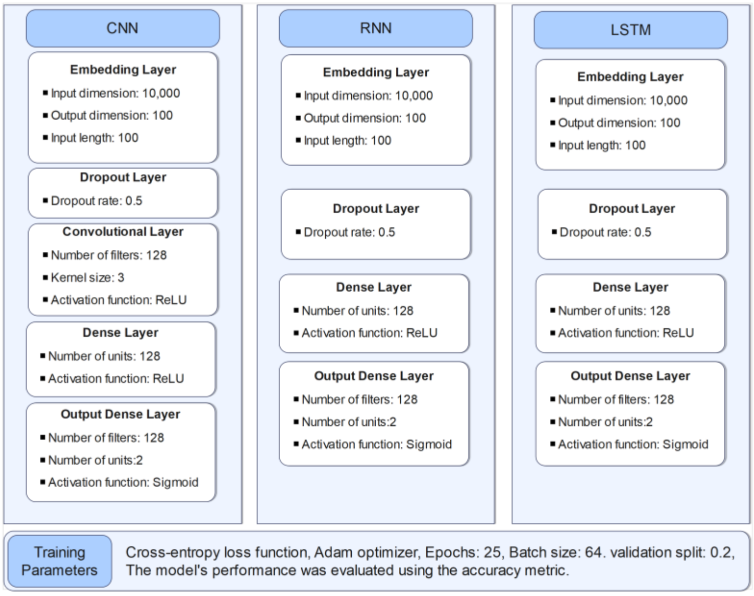

Figure 5. Classification Architecture for All Classifiers

The suggested model's accuracy is assessed based on the confusion matrix, which is a commonly used visual tool for demonstrating the performance of classification algorithms, which compares the correctly and incorrectly categorized values to the actual outcomes in the test data [65]. The accuracy assessment takes into account four variables, which are represented by a particular formula:

$$\boldsymbol{Accuracy = (TP + TN)/\,(TP + TN + FP + FN)} \qquad \textbf{… (2)}$$

Where:

$\boldsymbol{TP}$: are the cases that were correctly predicted by the model.
$\boldsymbol{TN}$ are the cases that were expected to be incorrect by the model.
$\boldsymbol{FP}$ are the cases predicted by the model but are incorrect.
$\boldsymbol{FN}$: These cases were predicted to be incorrect by the model but are true.

Python programming tools were used to carry out all of the study processes. The cleaned data has been divided into two sets: the training set, which comprised 80% of the total, and the testing set, which contained the rest of the data. The settings are identical for all the categorization and stemming tools and techniques.

The first experiment is conducted on the dataset using the CNN technique. As stated earlier, the classification process is performed on different processed data. Table 3 illustrates the results of the three classifiers (CNN, LSTM, RNN) along with different stemming techniques (Khoja, Snowball, Tashaphyne) also three FE methods (TF-IDF, N-Gram, BoW). The classification performance is described for each method so that Table 3 states the results according to the text processing methods. All three stemming techniques are evaluated according to the FE method used, in addition to the (No FE), which refers to the classification without using any feature extraction.

Table 3. Model Classification Performance

Classifiers	Stemming Method	Feature Extraction			
		No FE	TF-IDF	N-gram	BoW
CNN	No Stemming	0.923	0.913	0.924	0.915
	Khoja	0.927	0.910	0.932	0.914
	Snowball	0.929	0.914	0.935	0.917
	Tashaphyne	0.926	0.910	0.929	0.913
LSTM	No Stemming	0.928	0.909	0.915	0.911
	Khoja	0.921	0.907	0.914	0.908
	Snowball	0.929	0.913	0.918	0.915
	Tashaphyne	0.927	0.908	0.913	0.910

RNN	**No Stemming**	0.895	0.910	0.915	0.909
	Khoja	0.902	0.907	0.916	0.910
	Snowball	0.829	0.913	0.917	0.915
	Tashaphyne	0.907	0.905	0.913	0.909

From the results given, it is clearly can be seen that there are slight differences among the methods used. The best performance resulted from using (CNN+ Snowball+ N-Gram) with an accuracy of (0.935). At the same time, the lowest comes from using (RNN+ Snowball+ No FE) at (0.829) (Figure 6 & Figure 7). For the CNN classifier, the best performance is gained when using the Snowball stemmer along with N-Gram. It is noteworthy that using (N-Gram) feature can overcome other FE methods when using stemmer, which means that whenever stemmer is used, N-Gram can perform well. Despite that, N-Gram can give the best performance for our model, but we can see the model has some stability when dealing with the test when no FE is used. In other words, the performance of (No FE) ranges from (0.921) for (Tashaphyne) stemmer up to (0.927) for (No Stemming), which indicates low variance among various methods. While the best performance was gained when using (N-Gram), the performance has some fluctuation ranging from (0.913) for (No Stemming) up to (0.935) for (Snowball). According to our model, (N-Gram) is positively affected by the stemmer type, but when no FE method is used, it is best not to utilize stemmers since they have a minor impact on the results.

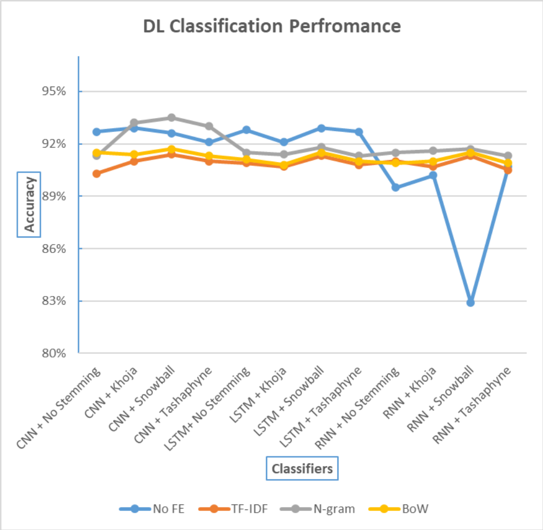

Figure 6. Classification Performance Across the Methods Used

For the LSTM classifier, the best performance comes from using a Snowball stemmer with no FE with an accuracy of (0.929), and the lowest performance goes to the (Khoja) stemmer and (TF-IDF) as FE at (0.907). According to our model, we notice that the LSTM performs better when no FE is used. This means FE reduces the classification accuracy slightly. Despite LSTM's performance being less than CNN, if we look at the results table, we notice that the LSTM method performs more stable compared to CNN, so there is no significant variety of overall methods used. The RNN classifier may seem to have the lowest performance compared to other classifiers, where there is a drop in the performance when using (Snowball+No FE) with an accuracy of (0.829). The highest performance for the RNN classifier comes from (Snowball+N-Gram). Here, we notice that the highest and lowest performance comes from using the same stemmer, but the critical role was related to the FE. This means RNN has a high sensitivity to the FE, which has a significant impact on the results according to our model.

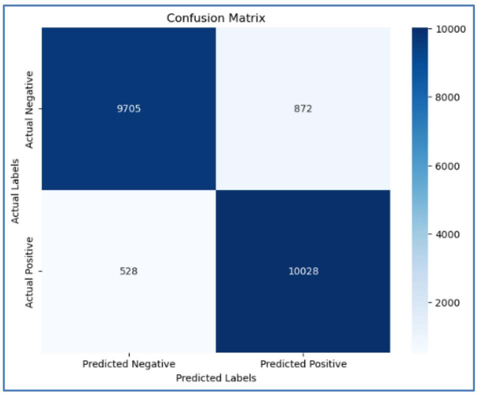

Figure 7. Confusion Matrix for the Best Performance

Regarding the performance of stemmers, the Khoja stemmer has a tiny level of variance among different classifiers and FE. Snowball performs very similarly in CNN and LSTM. With RNN, it has some variance, especially with No FE. Tashaphyne stemmer, on the other hand, has a good level of stability across the three classifiers with less sensitivity to FE methods. This means when using different classifiers, it is essential to test the performance of these classifiers along with stemming tools.

From the literature, the closest work to this study is presented by [28]; the authors used the HARD dataset and implemented an evaluation using different stemmers. The authors implemented different ML techniques (SVM, NB, LR, KNN), and the best performance was gained by using (LR + Snowball) and (SVM+ Snowball) with an accuracy of (91%). The current study differs from the past one in two things; first, the current study relies on implementing DL classifiers. Secondly, the current study involves using FE and evaluating the performance. The best performance gained when using DL, so our new study surpasses the past one at an accuracy of (93.5%), Table 4.

Table 5. Compare the Model Performance with the Literature

Literature	Description	Methods	Accuracy
El Rifai et al. [26]	Text categorization	CNN-GRU	94.85%
Khelil et al. [28]	Review Classification	ML techniques	91%
Najadat et al. [48]	Facebook Comments Classification	Decision Tree	92.63%
Our study	Review Classification	DL	93.5%

To compare our model to other works, Table 6 illustrates a study presented in [26] that involved the implementation of text categorization using several ML methods. The key difference between this study and ours is the dataset type, where the authors utilized a dataset of news articles, which in general are written in a formal style, while our dataset of freestyle, where most reviews are written dialectal manner. Except for the study presented in [26], our model presents promising results, which leads to pushing forward in investigating more techniques and methods toward digging deeply into similar approaches to present more solutions in this field.

5. CONCLUSION

In this study, we presented a model to classify Arabic reviews by utilizing DL classifiers along with different stemming and feature extraction techniques. Based on the results gained from the study's experiments, it is noteworthy to observe that the performance of Arabic text classification still needs more enhancement, and this is due to the Arabic text complexity. The performance of using different stemming techniques has a tiny impact on the performance results, and this is also related to the nature of the dataset, where there is a big part of the comments were written in a conversational style that has clear rules. Some methods are sensitive to using stemmers; others are not. FE, on the other hand, sometimes gives lower accuracy, which means that the text may lose some essential features when manipulating it using FE. The process of stemming can also affect the text quality and cause some features to be lost, and this explains some fluctuations in the results. Despite all the mentioned limitations, our model presented significant results by implementing widespread techniques of FE and stemming. The best performance of our model was gained by combining (CNN+ Snowball+ N-Gram) with an accuracy of (93.5%). The results of this study open the doors toward investigating more methods and techniques that deal with Arabic text classification. The implementation of word embedding methods might overcome the dialectal limitation by building some corpora for some Arabic dialects, which can serve in presenting more advanced models.

REFERENCES

[1] M. M. Almanea, “Automatic Methods and Neural Networks in Arabic Texts Diacritization: A Comprehensive Survey,” *IEEE Access*, vol. 9, no. Dl, pp. 145012–145032, 2021, doi: 10.1109/ACCESS.2021.3122977.

[2] O. Oueslati, E. Cambria, M. Ben HajHmida, and H. Ounelli, “A review of sentiment analysis research in Arabic language,” *Future Generation Computer Systems*, vol. 112, 2020, doi: 10.1016/j.future.2020.05.034.

[3] M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari, and A. Hilal, “Preprocessing Arabic text on social media,” *Heliyon*, vol. 7, no. 2, 2021, doi: 10.1016/j.heliyon.2021.e06191.

[4] T. K. Yeow and K. H. Gan, “Improving Comparative Opinion Mining Through Detection of Support Sentences,” 2022. doi: 10.4018/978-1-7998-9594-7.ch004.

[5] R. Kibble, “Introduction to natural language processing Undergraduate study in Computing and related programmes,” *Roeper Rev*, vol. 1, no. 2, 2013.

[6] M. B. Ressan and R. F. Hassan, “Naïve-Bayes family for sentiment analysis during COVID-19 pandemic and classification tweets,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 1, 2022, doi: 10.11591/ijeecs.v28.i1.pp375-383.

[7] S. L. Marie-Sainte, B. S. Alnamlah, N. F. Alkassim, and S. Y. Alshathry, “A new system for Arabic recitation using speech recognition and Jaro Winkler algorithm,” *Kuwait Journal of Science*, vol. 49, no. 1, 2022.

[8] I. Guellil, H. Saâdane, F. Azouaou, B. Gueni, and D. Nouvel, “Arabic natural language processing: An overview,” *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 5, pp. 497–507, 2021.

[9] H. Elzayady, M. S. Mohamed, K. M. Badran, and G. I. Salama, “Detecting Arabic textual threats in social media using artificial intelligence: An overview,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 3, pp. 1712–1722, 2022.

[10] X. Deng, Y. Li, J. Weng, and J. Zhang, “Feature selection for text classification: A review,” *Multimed Tools Appl*, vol. 78, no. 3, 2019, doi: 10.1007/s11042-018-6083-5.

[11] Ahmed Burhan Mohammed, “Decision Tree, Naïve Bayes and Support Vector Machine Applying on Social Media Usage in NYC / Comparative Analysis,” *Tikrit Journal of Pure Science*, vol. 22, no. 9, 2023, doi: 10.25130/tjps.v22i9.881.

[12] M. F. Ibrahim, M. A. Alhakeem, and N. A. Fadhil, “Evaluation of Naïve Bayes Classification in Arabic Short Text Classification,” *Al-Mustansiriyah Journal of Science*, vol. 32, no. 4, pp. 42–50, Nov. 2021, doi: 10.23851/mjs.v32i4.994.

[13] M. E. M. Abo *et al.*, “A multi-criteria approach for arabic dialect sentiment analysis for online reviews: Exploiting optimal machine learning algorithm selection,” *Sustainability (Switzerland)*, vol. 13, no. 18, 2021, doi: 10.3390/su131810018.

[14] N. T. Mohammed, E. A. Mohammed, and H. H. Hussein, “Evaluating Various Classifiers for Iraqi Dialectic Sentiment Analysis,” in *Lecture Notes in Networks and Systems*, 2023. doi: 10.1007/978-981-19-1412-6_6.

[15] A. Karimi, L. Rossi, and A. Prati, “AEDA: An Easier Data Augmentation Technique for Text Classification,” *Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*, pp. 2748–2754, 2021, doi: 10.18653/v1/2021.findings-emnlp.234.

[16] H. A. Almuzaini and A. M. Azmi, “Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization,” *IEEE Access*, vol. 8, pp. 127913–127928, 2020, doi: 10.1109/ACCESS.2020.3009217.

[17] L. Zhang, W. Jiang, and Z. Zhao, “Short-text feature expansion and classification based on non-negative matrix factorization,” in *Machine Learning for Cyber Security: Third International Conference, ML4CS 2020, Guangzhou, China, October 8–10, 2020, Proceedings, Part III 3*, Springer, 2020, pp. 347–362.

[18] H. Elzayady, K. M. Badran, and G. I. Salama, “Arabic Opinion Mining Using Combined CNN - LSTM Models,” *International Journal of Intelligent Systems and Applications*, vol. 12, no. 4, pp. 25–36, 2020, doi: 10.5815/ijisa.2020.04.03.

[19] A. M. Bdeir and F. Ibrahim, “A framework for arabic tweets multi-label classification using word embedding and neural networks algorithms,” in *Proceedings of the 2020 2nd International Conference on Big Data Engineering*, 2020, pp. 105–112.

[20] G. Lu, J. Gan, J. Yin, Z. Luo, B. Li, and X. Zhao, “Multi-task learning using a hybrid representation for text classification,” *Neural Comput Appl*, vol. 32, no. 11, pp. 6467–6480, 2020, doi: 10.1007/s00521-018-3934-y.

[21] W. Cherif, A. Madani, and M. Kissi, “Text categorization based on a new classification by thresholds,” *Progress in Artificial Intelligence*, vol. 10, no. 4, pp. 433–447, 2021, doi: 10.1007/s13748-021-00247-1.

[22] M. M. Saeed and Z. Al Aghbari, “ARTC: feature selection using association rules for text classification,” *Neural Comput Appl*, vol. 34, no. 24, pp. 22519–22529, 2022, doi: 10.1007/s00521-022-07669-5.

[23] S. M. Alzanin, A. M. Azmi, and H. A. Aboalsamh, "Short text classification for Arabic social media tweets," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6595–6604, 2022.

[24] S. K. Prabhakar, "Models with Multihead Attention," vol. 2021, 2021.

[25] A. Elnagar, Y. S. Khalifa, and A. Einea, "Hotel Arabic-reviews dataset construction for sentiment analysis applications," *Intelligent natural language processing: Trends and applications*, pp. 35–52, 2018.

[26] H. El Rifai, L. Al Qadi, and A. Elnagar, "Arabic text classification: the need for multi-labeling systems," *Neural Comput Appl*, vol. 34, no. 2, 2022, doi: 10.1007/s00521-021-06390-z.

[27] Y. S. and E. A. Elnagar Ashraf and Khalifa, "Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications," in *Intelligent Natural Language Processing: Trends and Applications*, A. E. and T. F. Shaalan Khaled and Hassanien, Ed., Cham: Springer International Publishing, 2018, pp. 35–52. doi: 10.1007/978-3-319-67056-0_3.

[28] Hawraa Fadhil Khelil, Mohammed Fadhil Ibrahim, Hafsa Ataallah Hussein, and Raed Kamil Naser, "Evaluation of Different Stemming Techniques on Arabic Customer Reviews," *Journal of Techniques*, vol. 6, no. 1, pp. 103–111, Feb. 2024, doi: 10.51173/jt.v6i1.2313.

[29] S. Alyami, A. Alhothali, and A. Jamal, "Systematic literature review of Arabic aspect-based sentiment analysis," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6524–6551, 2022.

[30] O. Oueslati, E. Cambria, M. Ben HajHmida, and H. Ounelli, "A review of sentiment analysis research in Arabic language," *Future Generation Computer Systems*, vol. 112, pp. 408–430, 2020.

[31] M. Alhanjouri, "Pre Processing Techniques for Arabic Documents Clustering," *International Journal of Engineering and Management Research*, no. 2, pp. 70–79, 2017.

[32] B. Jurish and K.-M. Würzner, "Word and Sentence Tokenization with Hidden Markov Models," *Journal for Language Technology and Computational Linguistics*, vol. 28, no. 2, pp. 61–83, 2013, doi: 10.21248/jlcl.28.2013.176.

[33] I. A. El-Khair, "Effects of Stop Words Elimination for Arabic Information Retrieval: A Comparative Study," pp. 1–15, 2017.

[34] A. Alajmi, E. M. Saad, and R. R. Darwish, "Toward an ARABIC stop-words list generation," *Int J Comput Appl*, vol. 46, no. 8, pp. 8–13, 2012.

[35] T. Kanan, O. Sadaqa, A. Almhirat, and E. Kanan, "Arabic light stemming: A comparative study between p-stemmer, khoja stemmer, and light10 stemmer," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, IEEE, 2019, pp. 511–515.

[36] K. Tan, C.-P. Lee, K. Lim, and K. Anbananthen, "Sentiment Analysis With Ensemble Hybrid Deep Learning Model," *IEEE Access*, vol. PP, p. 1, Jan. 2022, doi: 10.1109/ACCESS.2022.3210182.

[37] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Improving Sentiment Analysis in Arabic Using Word Representation," in *2nd IEEE International Workshop on Arabic and Derived Script Analysis and Recognition, ASAR 2018*, IEEE, 2018, pp. 13–18. doi: 10.1109/ASAR.2018.8480191.

[38] T. Kanan, O. Sadaqa, A. Almhirat, and E. Kanan, "Arabic Light Stemming: A Comparative Study between P-Stemmer, Khoja Stemmer, and Light10 Stemmer," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, IEEE, Oct. 2019, pp. 511–515. doi: 10.1109/SNAMS.2019.8931842.

[39] K. Abainia, S. Ouamour, and H. Sayoud, "A novel robust Arabic light stemmer," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 29, no. 3, pp. 557–573, May 2017, doi: 10.1080/0952813X.2016.1212100.

[40] F. E. Zamani, K. Umam, W. D. I. Azis, and W. S. Abdillah, "Analysis and implementation of computer-based system development of stemming algorithm for finding Arabic root word," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Dec. 2019. doi: 10.1088/1742-6596/1402/6/066030.

[41] [41] M. El-Masri, N. Altrabsheh, and H. Mansour, "Successes and challenges of Arabic sentiment analysis research: a literature review," *Soc Netw Anal Min*, vol. 7, no. 1, p. 54, Dec. 2017, doi: 10.1007/s13278-017-0474-x.

[42] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz, and A. Dahou, "A Study of the Effects of Stemming Strategies on Arabic Document Classification," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2903331.

[43] M. O. Alhawarat, H. Abdeljaber, and A. Hilal, "Effect of stemming on text similarity for Arabic language at sentence level," *PeerJ Comput Sci*, vol. 7, p. e530, May 2021, doi: 10.7717/peerj-cs.530.

[44] A. Oussous, A. A. Lahcen, and S. Belfkih, "Impact of Text Pre-processing and Ensemble Learning on Arabic Sentiment Analysis," *Proceedings of the 2nd International Conference on Networking, Information Systems \& Security*, 2019.

[45] X. Li, Z. Li, H. Qiu, G. Hou, and P. Fan, "An overview of hyperspectral image feature extraction, classification methods and the methods based on small samples," *Applied Spectroscopy Reviews*, vol. 58, no. 6. 2023. doi: 10.1080/05704928.2021.1999252.

[46] M. Avinash and E. Sivasankar, "A study of feature extraction techniques for sentiment analysis," in *Advances in Intelligent Systems and Computing*, 2019. doi: 10.1007/978-981-13-1501-5_41.

[47] X. Chen, Y. Xue, H. Zhao, X. Lu, X. Hu, and Z. Ma, "A novel feature extraction methodology for sentiment analysis of product reviews," *Neural Comput Appl*, vol. 31, pp. 6625–6642, 2019.

[48] H. Najadat, M. A. Alzubaidi, and I. Qarqaz, "Detecting Arabic Spam Reviews in Social Networks Based on Classification Algorithms," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 1. Association for Computing Machinery, Jan. 01, 2022. doi: 10.1145/3476115.

[49] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on the sentiment analysis," *Procedia Comput Sci*, vol. 152, pp. 341–348, 2019.

[50] A. Madasu and S. Elango, "Efficient feature selection techniques for sentiment analysis," *Multimed Tools Appl*, vol. 79, no. 9–10, pp. 6313–6335, 2020, doi: 10.1007/s11042-019-08409-z.

[51] J. Mutinda, W. Mwangi, and G. Okeyo, "Lexicon-pointed hybrid N-gram Features Extraction Model (LeNFEM) for sentence level sentiment analysis," *Engineering Reports*, vol. 3, no. 8, 2021, doi: 10.1002/eng2.12374.

[52] F. Shannag, B. H. Hammo, and H. Faris, "The design, construction and evaluation of annotated Arabic cyberbullying corpus," *Educ Inf Technol (Dordr)*, vol. 27, no. 8, pp. 10977–11023, Sep. 2022, doi: 10.1007/s10639-022-11056-x.

[53] M. Alhawarat and A. O. Aseeri, "A Superior Arabic Text Categorization Deep Model (SATCDM)," *IEEE Access*, vol. 8, pp. 24653–24661, 2020, doi: 10.1109/ACCESS.2020.2970504.

[54] L. Zhang, W. Jiang, and Z. Zhao, "Short-text feature expansion and classification based on nonnegative matrix factorization," *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 10066–10080, 2022, doi: 10.1002/int.22290.

[55] S. Larabi-Marie-Sainte, B. S. Alnamlah, N. F. Alkassim, and S. Y. Alshathry, "A new framework for Arabic recitation using speech recognition and the Jaro Winkler algorithm," *Kuwait Journal of Science*, vol. 49, no. 1, 2022, doi: 10.48129/KJS.V49I1.11231.

[56] S. Boukil, M. Biniz, F. El Adnani, L. Cherrat, and A. E. El Moutaouakkil, "Arabic text classification using deep learning technics," *International Journal of Grid and Distributed Computing*, vol. 11, no. 9, pp. 103–114, 2018, doi: 10.14257/ijgdc.2018.11.9.09.

[57] T. Kanan and E. A. Fox, "Automated arabic text classification with P-S temmer, machine learning, and a tailored news article taxonomy," *J Assoc Inf Sci Technol*, vol. 67, no. 11, pp. 2667–2683, 2016.

[58] W. Alabbas, H. M. Al-Khateeb, and A. Mansour, "Arabic text classification methods: Systematic literature review of primary studies," *Colloquium in Information Science and Technology, CIST*, vol. 0, no. x, pp. 361–367, 2016, doi: 10.1109/CIST.2016.7805072.

[59] S. Bodapati, H. Bandarupally, R. N. Shaw, and A. Ghosh, "Comparison and analysis of RNN-LSTMs and CNNs for social reviews classification," *Advances in Applications of Data-Driven Computing*, pp. 49–59, 2021.

[60] D. Alsaleh and S. Larabi-Marie-Sainte, "Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms," *IEEE Access*, vol. 9, pp. 91670–91685, 2021, doi: 10.1109/ACCESS.2021.3091376.

[61] B. Jang, M. Kim, G. Harerimana, S. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," *Applied Sciences*, vol. 10, no. 17, p. 5841, 2020.

[62] M. Ahmed, P. Chakraborty, and T. Choudhury, "Bangla document categorization using deep RNN model with attention mechanism," in *Cyber Intelligence and Information Retrieval: Proceedings of CIIR 2021*, Springer, 2022, pp. 137–147.

[63] J. Du, C.-M. Vong, and C. L. P. Chen, "Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification," *IEEE Trans Cybern*, vol. 51, no. 3, pp. 1586–1597, 2020.

[64] X. Li and H. Ning, "Ce text classification based on hybrid model of CNN and LSTMhines," in *Proceedings of the 3rd International Conference on Data Science and Information Technology*, 2020, pp. 129–134.

[65] J. Han, J. Pei, and H. Tong, *Data mining: concepts and techniques*. Morgan kaufmann, 2022.