applied sciences

MDPI

Article

An Improved Sentiment Classification Approach for Measuring User Satisfaction toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique

Mohammed Hadwan [1,2,*], Mohammed Al-Sarem [3,4], Faisal Saeed [3,5] and Mohammed A. Al-Hagery [6]

[1] Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[2] Department of Computer Science, College of Applied Sciences, Taiz University, Taiz 6803, Yemen
[3] College of Computer Science and Engineering, Taibah University, Medina 42353, Saudi Arabia; msarem@taibahu.edu.sa (M.A.-S.); fsaeed@taibahu.edu.sa (F.S.)
[4] Department of Computer Science, Saba'a Region University, Mareb, Yemen
[5] DAAI Research Group, Department of Computing and Data Science, School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK
[6] Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia; hajry@qu.edu.sa
* Correspondence: m.hadwan@qu.edu.sa

check for updates

Citation: Hadwan, M.; Al-Sarem, M.; Saeed, F.; Al-Hagery, M.A. An Improved Sentiment Classification Approach for Measuring User Satisfaction toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique. *Appl. Sci.* **2022**, *12*, 5547. https://doi.org/10.3390/app12115547

Academic Editors: Carlos A. Iglesias and Antonio Moreno

Received: 19 April 2022
Accepted: 27 May 2022
Published: 30 May 2022

Abstract: Analyzing the sentiment of Arabic texts is still a big research challenge due to the special characteristics and complexity of the Arabic language. Few studies have been conducted on Arabic sentiment analysis (ASA) compared to English or other Latin languages. In addition, most of the existing studies on ASA analyzed datasets collected from Twitter. However, little attention was given to the huge amounts of reviews for governmental or commercial mobile applications on Google Play or the App Store. For instance, the government of Saudi Arabia developed several mobile applications in healthcare, education, and other sectors as a response to the COVID-19 pandemic. To address this gap, this paper aims to analyze the users' opinions of six applications in the healthcare sector. An improved sentiment classification approach was proposed for measuring user satisfaction toward governmental services' mobile apps using machine learning models with different preprocessing methods. The Arb-AppsReview dataset was collected from the reviews of these six mobile applications available on Google Play and the App Store, which includes 51k reviews. Then, several feature engineering approaches were applied, which include Bing Liu lexicon, AFINN, and MPQA Subjectivity Lexicon, bag of words (BoW), term frequency-inverse document frequency (TF-IDF), and the Google pre-trained Word2Vec. Additionally, the SMOTE technique was applied as a balancing technique on this dataset. Then, five ML models were applied to classify the sentiment opinions. The experimental results showed that the highest accuracy score (94.38%) was obtained by applying a support vector machine (SVM) using the SMOTE technique with all concatenated features.

Keywords: sentiment analysis; machine learning; classification; ensemble learning; feature selection

1. Introduction

Globally, the number of mobile application users has increased dramatically. Besides that, many social networking sites have been developed allowing people to express their thoughts regarding controversial events or matters. This helps to increase the number of content creators who utilize social networking media such as Twitter, Facebook, and others. In addition, the users of these mobile applications can add reviews using different platforms such as Google Play and the App Store. These platforms allow users to be a part of content creators by posting comments regarding the content and quality of these applications. Due to the huge expansion of the content posted on these sites daily, institutions including government agents and private companies have exploited people's expressions and opinions regarding the services or products that are provided online.

Sentiment analysis is the task of utilizing text mining, natural language processing (NLP), and computational linguistics approaches to systematically detect, analyze, and examine the users' opinions appearing in subjective textual information. Sentiment analysis processes concentrate on detecting text containing opinions and deciding whether these opinions are positive, negative, or neutral comments [1,2]. Recently, sentiment analysis approaches have relied on inspecting opinions or emotions of diverse subjects, such as peoples' opinions and impressions about movies, products, and daily affairs.

In the literature, several researchers have utilized supervised machine learning algorithms, especially classification methods, for sentiment analysis purposes, such as support vector machine (SVM) and naïve Bayes classifiers. Alomari [3] showed that the SVM classifier with both TF-IDF and stemming outperformed the naïve Bayes classifier for Arabic tweets' sentiment analysis. Similarly, Abuelenin et al. [4] combined machine learning classifiers, ISRI Arabic stemmer, and the cosine similarity to propose a better-performing hybrid method. The experimental results in [4] demonstrated that the ISRI Arabic stemmer with linear SVM can achieve higher accuracy compared to other stemmers, such as the Porter stemmer. In addition, Shoukry and Rafea [5] used SVM and naïve Bayes classification algorithms for sentiment analysis. Unigram and bigram features were extracted to train the classifiers. The outcomes showed that SVM performed better than NB on the collected tweets using Twitter API.

On the other hand, deep neural networks (DNN) have been applied and shown a good performance compared to traditional machine learning in terms of detecting sentiments of short texts. A number of sentiment analysis studies have showed that convolution neural networks (CNN) and long–short-term memory networks (LSTM) [6] outperform the traditional machine learning approaches to detect sentiment. In addition, combining LSTM with CNN [7] showed promising results and surpassed the performance of the conventional machine learning models.

Several research works were conducted for Arabic sentiment analysis (ASA) that gained more interest recently, especially during the COVID-19 pandemic. According to [8], there are three main approaches used for Arabic sentiment analysis, which are supervised, unsupervised, and hybrid methods. The research conducted obtained interesting outcomes, but at the same time, the results were more divergent because of the different types of methods used and Latin languages, where only a few research works were conducted for studying ASA, which focused on analyzing Arabic tweets. Therefore, more efforts are still needed to address the sentiment analysis of users' reviews on mobile applications, especially the applications that provide governmental services in health, education, and other sectors. Recently, different mobile applications were developed in Saudi Arabia after the COVID-19 pandemic. According to the World Health Organization (WHO), Saudi Arabia actively participated in fighting COVID-19 nationally, regionally, and globally. Locally, the government has taken several urgent actions to fight COVID-19 in different sectors, such as health, education, security, Islamic affairs, and others. Several mobile applications were developed to provide online services. For instance, these healthcare applications, namely Tawakkalna, Tetaman, Tabaud, Sehhaty, Mawid, and Sehhah, were successfully launched and used by millions of users in Saudi Arabia during the COVID-19 pandemic. This study will review the most recent studies in this field and propose a sentiment classification approach for measuring user satisfaction toward governmental services' mobile applications. This analysis will support the government officers to make better decisions regarding the improvements in the quality of the online services offered to citizens and residents. In addition, the paper will help the developers to improve any potential bugs or difficulties in these applications based on the users' opinions and experiences.

The key contributions of this research paper can be summarized as follows:

- This study presents the first use of the Arb-AppsReview dataset that is designed to capture users' reviews on the Google Play Store. The original dataset is available publicly on github.com at (https://github.com/Arb-AppsReview/Arb-AppsReview/blob/main/apps_ar_reviews_dataset.csv) (accessed on 20 November 2021). The dataset comprises about 51,000 Arabic reviews related to six Saudi governmental services' apps.
- The original dataset finds the sentiment score for each review by running the Camel tool (https://camel-tools.readthedocs.io/en/latest/api/sentiment.html) (accessed on 1 March 2022). However, the authors found that some reviews were labeled incorrectly. Thus, the dataset was enriched by several lexical dictionaries. In this study, the Arabic TextBlob lexical dictionary was integrated firstly for annotating the users' reviews, and later the performance of the ML classifiers was compared with the original dataset labeled by the Camel tool.
- Several feature engineering approaches, namely, Bing Liu lexicon, AFINN, and MPQA Subjectivity Lexicon, a bag of words (BoW), term frequency-inverse document frequency (TF-IDF), and the Google pre-trained Word2Vec, were integrated.
- Several experiments were carried out in this study to compare the performance of the proposed feature extraction techniques using five ML models, including random forest (RF), bagging, support vector machine (SVM), logistic regression (LR), and naïve Bayes (NB).
- The performance of the selected ML models was first investigated using an imbalanced dataset. Later, further experiments were performed using a balanced dataset. As balancing techniques, both under-sampling and oversampling were used. In this regard, the SMOTE technique was applied as a balancing technique on this dataset and obtained better enhancements.

The rest of the paper is organized as follows: Section 2 presents the recent studies on Arabic sentiment analysis. The materials and methods are presented in Section 3, which briefly describes the dataset along with the used preprocessing techniques. In addition, it explains the techniques and algorithms utilized in this research. In Section 4, the results and discussions of the proposed approach are highlighted. Finally, Section 5 concludes the whole paper.

2. Related Works

The Arabic language has special characteristics that add additional challenges when addressing the Arabic sentiment analysis. These challenges include morphological analysis, dialectal Arabic, Arabizi, and named entity recognition, in addition to the Arabic structure complexity and having different cultures [8,9]. This makes it necessary to conduct more research and propose new methods for enhancing the sentiment analysis in Arabic. More efforts were made to analyze data from Twitter for serval purposes, such as analyzing the users' opinions about online learning during COVID-19. This section focuses on reviewing the existing studies that used machine leaching and deep learning for Arabic sentiment analysis. More focus will be given to the studies that were applied for addressing the sentiment analysis of the Saudi Arabia community.

According to [10], there are many benefits to analyzing Arabic sentiment, such as showing valuable insights for different provided services [11], recognizing the potential influencers in social media [12], and email spam detection [13]. There are three main strategies with different challenges for ASA, and these include preprocessing, feature generation, and selection and classification methods. Several studies proposed different methods for each of these strategies to enhance the performance of ASA. For instance, applying different preprocessing methods such as stemming, tokenization, and normalization can improve the performance of the sentiment analysis [10]. In addition, the effect of stemming on the ASA problem was studied in [14]. They used the Arabic root stemmer of Khoja and the light stemmer on two datasets and found that the latter performed better for sentiment classification.

Other studies highlighted the importance of data preprocessing for Arabic sentiment analysis. For instance, the authors of [15] recommended that the preprocessing tools such as normalization, tokenization, and stop words' removal should be utilized in all ASA studies to improve the performance of the analysis. Additionally, they recommended ending the preprocessing by applying stemming methods, although aggressive stemming was not recommended as it might change the Arabic words' polarity.

It is worth mentioning that most of the research efforts on the Arabic language aimed to discuss the modern standard form. However, by browsing the social media websites, we can find that majority of the Arabic users used their dialects, which generates a huge amount of Arabic dialects texts [16]. Different Arabic dialects were addressed, such as Saudi, Iraqi, Egyptian, Jordanian, Palestinian, and Algerian dialects [16]. According to [17], most of the studies focused on Jordanian (38%), Egyptian (23%), and Saudi (15%) dialects. For instance, Mustafa et al. [18] applied automatic extraction of opinions on social media that are written in modern standard Arabic and Egyptian dialects, and they automatically analyzed sentiment into either positive or negative. Gamal et al. [19] applied ML methods for analyzing the sentiment of Arabic dialects. They applied different classifiers using a labeled dataset and found that the classifiers obtained good results using different evaluation metrics.

To address the challenges of the Arabic language, Touahri and Mazroui [20] created both stemmed and lemmatized versions of word lexicons for integrating the morphological notion. Then, a supervised model was constructed from a set of features. In addition, they semantically segmented the lexicon for reducing the vector's size of the model and enhancing the execution time. In addition, Aloqaily et al. [21] proposed lexicon-based and ML methods for sentiment analysis. They used the Arabic tweets dataset. The outcomes showed that ML methods, especially logistic model trees and SVM, outperformed the lexicon-based methods in predicting the subjectivity of tweets. The outcomes also showed the importance of applying feature selection for improving the performance of ASA.

Several studies were conducted on ASA in Saudi Arabia. For instance, Aljameel et al. [22] developed a prediction model for people's awareness of the precautionary procedures in Saudi Arabia. They used an Arabic COVID-19-related dataset that was generated from Twitter. Three predictive models were applied, which included SVM, K-NN, and naïve Bayes, with the N-gram feature extraction method. The experimental results showed that SVM with bigram in TF-IDF outperformed other methods, and obtained 85% of prediction accuracy. The applied method was recommended to help the decision-makers in the medical sectors to apply different procedures in each region in Saudi Arabia during the pandemic. In addition, the authors of [23] studied the attitude of individuals in Saudi Arabia about online learning. They collected Arabic tweets posted in 2020 and applied sentiment analysis to this dataset. The results showed that people have maintained a neutral response toward online learning. The authors of [24] also collected a dataset that includes 8144 tweets related to Qassim University in Saudi Arabia. They applied one-way analysis of variance (ANOVA) as a feature selection method for removing the irrelevant and redundant features for sentiment analysis of Arabic tweets. Then, the results showed that SVM and naïve Bayes achieved the best results with one-way ANOVA compared to other ML methods on the same dataset.

Deep learning was also applied to enhance the performance of Arabic sentiment analysis. Several studies utilized deep learning methods for this purpose; for instance, the authors of [25] used an ensemble model that combined convolutional neural network (SNN) and long–short-term memory (LSTM) methods for analyzing the sentiment of Arabic tweets. The proposed model in [25] achieved a 64.46% F1-score, which is considered higher than the applied deep learning methods on the same dataset. Moreover, the authors of [26] proposed a feature ensemble model of surface (manually extracted) and deep (sentiment-specific word embedding) features. The models were applied on three Arabic tweets datasets. The results showed that the proposed ensemble of surface and deep features models obtained the highest performance for sentiment analysis. In addition, Mohammed and Kora [27] proposed a corpus of 40,000 labeled Arabic tweets and applied 3 deep learning models for Arabic sentiment analysis, which were CNN, LSTM, and recurrent convolution neural network (RCNN). The results showed that LSTM outperformed CNN and RCNN with an average accuracy of 81.31%. Additionally, it was found that when data augmentation was applied to the corpus, the accuracy of LSTM increased by 8.3%. Another study on ASA [28] applied the multilayer, bidirectional, long–short-term memory (BiLSTM) method, that used the pre-trained word-embedding vectors. The applied model showed a notable enhancement in the performance of sentiment analysis compared to other models. The authors of [29] extracted Twitter data from different cities in Saudi Arabia. NLP and ML methods were used to analyze the sentiments of individuals during the COVID-19 pandemic. This research collected Arabic tweets, and then after manual annotation to classify the tweets into different sentiments, such as negative, positive, neutral, etc., they applied LSTM and naïve Bayes for classification. Similar to other studies, the results here showed that the LSTM model performed better than other models and obtained high accuracy. In [30], the performance of ML-based models, as well as the performance of deep learning models such as CNN, LSTM, CNN-LSTM, and Bi-LSTM, was examined using a set of 17,155 tweets about e-learning systems. The authors adopted TextBlob, VADER (Valence Aware Dictionary for Sentiment Reasoning), and SentiWordNet to analyze the polarity and subjectivity score of tweets' text. To ensure the quality of the dataset, the SMOTE technique was applied as a balancing technique for the dataset. The results showed that the TextBlob technique yielded the best accuracy of 94% when applied with the Bi-LSTM model. The performance of the CNN-LSTM model was also investigated in [31]. Although the model was tested on three non-Arabic datasets, the results demonstrated that combining CNN and LSTM is a good idea and produces a stable performance against the three datasets.

Referring to the studies conducted in the literature, we can obviously find that most of the studies were conducted using tweet datasets. This is maybe because the researchers prefer to use short text datasets. However, little attention was given to analyzing the huge amount of users' reviews for important mobile applications, such as governmental mobile apps. Therefore, this study investigates the analysis of sentiments for the Arabic dataset collected for six mobile apps available on the Google Play Store. The study also proposed a sentiment classification approach for measuring user satisfaction toward these governmental services' mobile apps.

3. Materials and Methods

This section presents the framework methodology, which was performed to achieve the study's objectives. The ASA was applied to analyze the users' reviews for six mobile apps that are providing some governmental services in Saudi Arabia. The scope of this study is limited to address the apps' reviews that are written in Arabic. To use lexical dictionaries which are not available in Arabic, the original reviews were translated first into English using Google translation APIs. The findings can be extended to other languages following the same framework. Figure 1 shows the proposed framework for conducting ASA at the word level.

According to Figure 1, the main steps are the dataset collection and annotation, pre-processing, feature extraction, feature selection, opinion classification, and performance measurements. In addition, the experimental part of this work was conducted to show the effectiveness of the proposed model using both a balanced and an imbalanced dataset. These steps are described in detail in the following subsections.

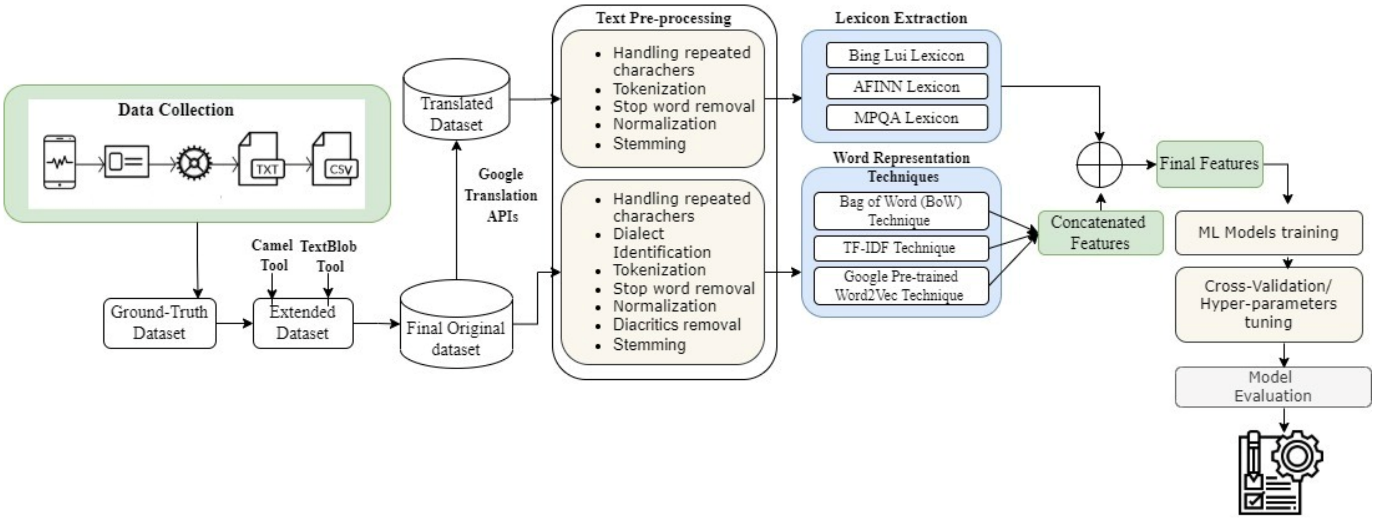

Figure 1. The study's overall framework.

3.1. Dataset Preparation

3.1.1. Dataset Description

The original dataset contains 51,767 user reviews which were obtained by scrapping six mobile apps available on Google Play Store. The content of the reviews was written in Arabic. For scrapping reviews, we used the Google Play Scrapper tool (https://github.com/JoMingyu/google-play-scraper) (accessed on 1 March 2022), which is designed to scrape the app contents using the app's ID found in the URL and the app name.

Since the Google Play Store allows for capturing information about apps under concern, the scrapper can collect, besides the users' reviews, some additional metadata about each review, such as the username, review date, the app version, thumbs up count, and the rating score.

Often, the scrapper returns reviews in textual format and saves all the collected information in comma-separated values (CSV) format. In addition, the distribution of reviews with respect to each app and some statistical information of Arb-AppsReview are presented in Table 1. Table 2 shows some associated metadata per app obtained directly from the Google Play Store. It is important to mention that these apps were classified as the most downloaded and used mobile apps in Saudi Arabia during the COVID-19 pandemic period.

Table 1. Statistics of the Arb-AppsReview dataset.

App Name	No. of Reviews of Arb-AppsReview	Distribution of Reviews (%)
Tawakkalna	21,009	41%
Tetaman	1356	3%
Tabaud	2369	5%
Sehhaty	4975	10%
Mawid	20,007	39%
Sehhah	2050	4%
Total number of Instances		51,767
Average word length (characters)		28.9
number of words per review (word)		4.998

Table 2. Google Play Store metadata of mobile apps.

App Name	# Downloads (Millions)	Rating	Current Version	Size (Mb)	Total No. of Reviews
Tawakkalna	10M	4.5	3.3.0	131	575,842
Tetaman	1M	2.7	1.7	26	6088
Tabaud	5M	4.6	1.2.0	6.6	23,992
Sehhaty	10M	4.4	2.13.2	42	263,004
Mawid	1M	4.7	10.10.0	61	126,941
Sehhah	1M	4.0	1.0.35	73	6121

3.1.2. Data Annotation

As stated earlier, the web scrapper provides us with the necessary information for each app, including the name of the user who wrote the review, the review itself, the date when the review was posted, thumbs up count, current version, and rating of the app. We noted that some users classify their review incorrectly and associate the review with a wrong score, which makes the annotation process based on this feature misleading. Table 3 presents a few samples of reviews from the dataset that show inconsistency in the given review and the score selected by the user. Furthermore, the “Rating” feature associated with the app metadata and provided by the Google Play Store is also not accurate enough. Thus, we added a new attribute called “Sentiment Polarity” to the dataset and annotated the reviews as positive, neutral, or negative sentiment. Thus, the final dataset consists of six features which are listed and described in Table 4. The next subsection shows how the “Sentiment Polarity” feature was added to the dataset.

Table 3. Review sample from the dataset with a misleading score assigned by a user.

No.	Review Content	Translated Review	Score
1	ليش ما يتحمل !!اله يومين يقلي جاري التثبيت	It cannot be installed!! Two days it said the installation is in progress	5
2	مارضى يتحمل	It cannot be installed	5
3	التطبيق متوقف لا يعمل في جوالي	The app is crashed and it does not work on my phone	5
4	جيد	Good	1
5	.ماشاء الله تطبيق ممتاز، يعطيكم العافية	Masha Allah, an excellent application, may God bless you	1

Table 4. Dataset description.

Features	Description
UserName	The name of the user who wrote the review
ReviewContent	Content of review
DateOfPost	The date when the review was posted
ThumbsUpCount	Number of users who have the same feelings
Rating	Five-scale rating represents a score given by the user (0 = low, 5 = high)
Lang	Language of review
SentimentPolarity	Sentiment label corresponding to each review (−1 = negative, 0 = neutral, 1 = positive)

3.1.3. Building Ground-Truth Dataset

Some users, as stated earlier, assigned rating scores for their reviews inconsistently. Thus, a new “Sentiment Polarity” feature was added to the dataset. This feature will be used as a sentiment class label used for training the ML classifiers. To avoid errors caused by users when they assign rating scores incorrectly, we asked three annotators to voluntarily judge 1135 reviews by giving, based on their opinions, a score for each review they revised.

As each annotator assigned a sentimental score separately, we followed the same guideline that was used in [32,33].

The final sentiment polarity class label was assigned by computing the majority voting algorithm and labeling the review as positive, neutral, or negative.

Figure 2 shows the distribution of polarity scores in the baseline dataset, which includes 522 instances that were labeled as positive reviews, 375 instances as negative reviews, and 238 instances as neutral reviews. Later, the Camel tool was used to predict the remaining instances in the dataset. To ensure that the tool assigned a correct sentiment polarity score for each review, the confession matrix was computed by calculating the values predicted by the Camel tool and those assigned by the annotators. The accuracy score shows that the Camel tool can predict class labels with an accuracy of 98.26%. Hence, employing the Camel tool on the remaining instances allowed us to extend the baseline dataset and the final distribution of sentiment polarity, as shown in Figure 3.

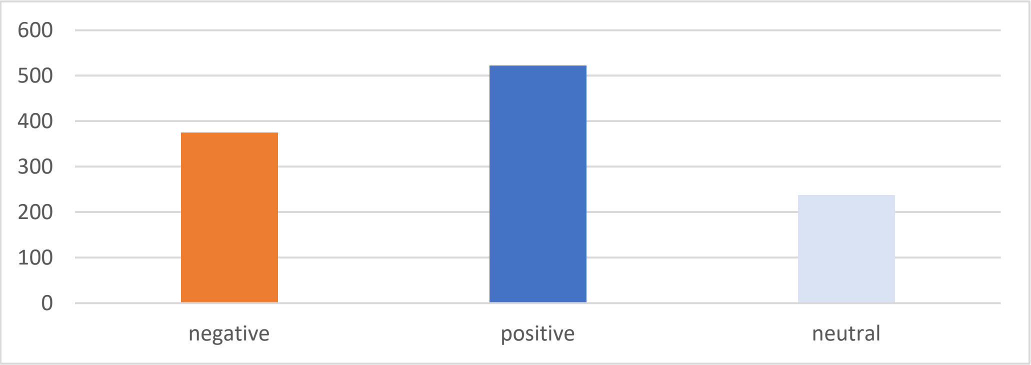

Figure 2. Distribution of sentiment score in the baseline dataset.

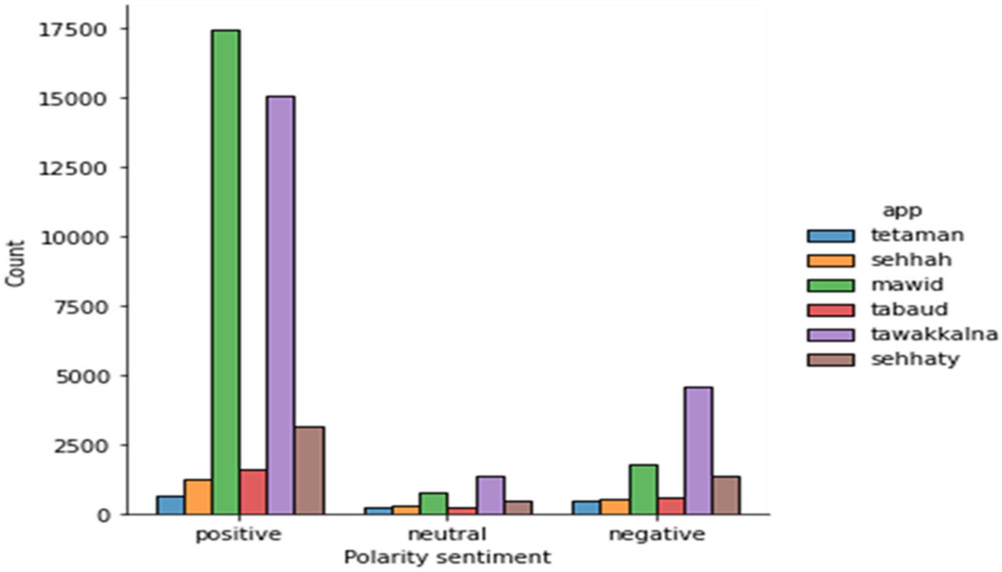

Figure 3. Distribution of sentiment for each app.

3.2. Data Preprocessing

Like any data extracted from online platforms, the reviews captured from the Google Play Store are largely unstructured. They might contain unnecessary data, which can negatively affect the performance of ML classifiers. This makes preprocessing of the data a very important step for improving the performance of the ML models and reducing the training time [32,34,35]. In addition, the size of the featured set can be reduced from 50% to 30%, as stated by the authors of [36].

An effective preprocessing method for the Arabic reviews should include data cleaning, spell checking for fixing any typo-errors in the posted text, and dialect identification. The data cleaning involves removing non-Arabic characters, punctuation, stop words, text normalization, dealing with diacritics, stemming, and handling words with repeating characters, which are described as follows.

3.2.1. Data Cleaning

The role of data cleaning is to remove unnecessary data which are insignificant for the analysis. The numbers and punctuation are a type of data that do not have an impact on the overall performance of ML classifiers, and they are irrelevant for the sentiment classification of text. Stop words are also a type of unnecessary data that add unnecessary complexity to the text analysis. Thus, we removed numbers, punctuation, non-Arabic characters, and stop words from the reviews.

Handling Words with Repeating Characters

We also noted that some users add some words with repeating characters. Although this type of repetition can be considered as a sign of emphasizing users' emotions, it also needs to be fixed before passing the data to the next preprocessing steps. Spelling check tools can be very useful for fixing such problems. However, we should be careful when the repeating characters are removed. The spelling checker not only has to handle the repeating characters in a word, but it also has to produce a word that is consistent with the overall sentence context, e.g., the word "كرر" (repeat) in the following sentence: "كرر المعلم الدرس" (The teacher repeats the lesson) contains two "ـر" characters. Therefore, if the spelling checker removes the repeated character (in this case the letter "ر"), a new word "كر" will be produced, which is absolutely a different word and it is irrelevant to the sentence context.

Dialect Identification

The Arabic users may write their review content freely using either the modern standard Arabic (MSA) or their "spoken" forms, each of which is a regional dialect (allahjah, "accent"). There are several regional dialects, such as Egyptian, Levantine, Gulf, Iraqi, Maghrebi, and Yemeni. Identifying to which region a user belongs improves the accuracy of the ASA systems by avoiding wrongly classifying a review content. Table 5 shows a review of content with different regional dialects.

Table 5. A sample of a review written by different users with different regional dialects.

Translated Review to English	Sample Review	Variant of Arabic
Unfortunately, what is a bad app. I've been trying to sign up for two days, no result.	.للأسف، يا لهُ من تطبيق سيئ. أحاول التسجيل منذ يومين دون فائدة.	MSA
	!!للأسف تطبيق مش قوي، لي يومين مشقادر اسجل	Yemeni
	!!وش ذا التطبيق السيئ، يومين ماقدرت اسجل	Gulf
	!!للأسف تطبيق سيئ جداً بقالي يومين بحاول اسجل	Egyptian
	.لك شو هالتطبيء السيء هاد. لك صارلي يومين .بدي سجل ومابحسن. هيك عالفاضي ومافي فايدى.	Levantine
	شلون تطبيق تعبان، صارلي يومين اريد اسجل وماكو نتيجة	Iraqi
	ما ستطعتش ندخل على الموقع من نهارين	Maghrebi

3.2.2. Normalization and Diacritics Removal

Text normalization is also an essential step that involves the transformation of a text into a standard form. Since most Arabic users write their online reviews without care for the grammar and dictation rules, several word forms might occur. In addition, Arabic inscription uses special characters (called Tashkeel, which is translated into English as diacritics) for producing several words from the same word root or different meanings. In this work, the following normalization rules are used:

- Normalize Hamza ء،ؤ، ـئ→ ء
- Normalize Alef أ→ إ,أ,ا
- Normalize Heh ـه ـة ، ـه→ هـ
- Normalize Caf ک→ ك
- Normalize Ye'a ي→ ئ ، ى
- Normalize lamalef لا لأ ، لآ→ لا

In addition, we remove all added vocalizations or diacritization on Arabic alphabets. For this purpose, the Tashaphyne Arabic light tool was used (https://pypi.org/project/Tashaphyne/) (accessed on 1 March 2022).

3.2.3. Stemming

Stemming involves the conversion of words into their root forms by deleting affixes from the words [24]. For instance, small, smaller, and smallest are variations of the root word "small" with the same meaning. By the process of stemming, the complexity of the textual feature is reduced, which enhances the learning ability of the classifier. A sample of reviews from the dataset before and after preprocessing is illustrated in Table 6.

Table 6. Sample tweets after preprocessing.

Sample Review	Preprocessed Review	Preprocessed Review in English
للأسف، يا لهُ من تطبيق سيئ. أحاول التسجيل منذ يومين دون فائدة	أسف يا ل من تطبيق سيئ. حاول تسجيل منذ يوم دون فائدة	nfortun, what is a bad app. i've been tri to sign up for two day, no result.
ما قدرت اسجل في التطبيق،. اوصل ل آخر خطوه وهي كود التحقق. كود التحقق ما يوصل على الجوال	ما قدر اسجل فس تطبيق وصل ل آخر خطو هي. كود تحقق كود تحقق ما وصل على جوال	I could not regist in the applic reach the last step, which is the verif code. the verif code is not sent to the mobil
جدا رائع الله يجزاكم خير الجز. اء وفرتوا علينا وقت وكل شي	جد رائع الله يجزاكم خير. جزاء فر على وقت كل شي	veri wonder, may god reward you with the best. you save us time and everyth

3.3. *Lexical Dictionaries*

In the following subsections, we briefly present the lexical dictionaries used in this research work. This section also shows how the proposed lexical dictionaries can be useful for improving the performance of the ML models.

3.3.1. TextBlob

TexBlob is a freely downloadable lexical dictionary that provides a simple API for performing natural language processing (NLP) tasks, such as ASA, part-of-speech tagging, classification, n-grams, noun phrase extraction, and more [37]. NaiveBayesAnalyzer and PatternAnalyzer are two modules that are integrated into the TexBlob sentiment module. The NaiveBayesAnalyzer module returns its result as a named tuple of the form <classification type, positive score, negative score>, whilst the PatternAnalyzer module returns the results as <polarity, subjectivity, assessments>. The sentiment scores have a range from +1.0 to −1.0, where +1.0 indicates positive sentiment and −1.0 indicates negative sentiment. To support Arabic sentiment analysis, we incorporated the TextBlob-ar 0.0.2 extension (https://github.com/adhaamehab/textblob-ar) (accessed on 1 March 2022).

After employing NaiveBayesAnalyzer, the annotation process is revised, and the dataset used is labeled with the value assigned in the classification type of NaiveBayesAnalyzer.

Comparing TextBlob's prediction of the sentiment of users' reviews with the original sentiment obtained earlier by using the Camel tool, Figure 4 shows the number of positive, neutral, and negative reviews.

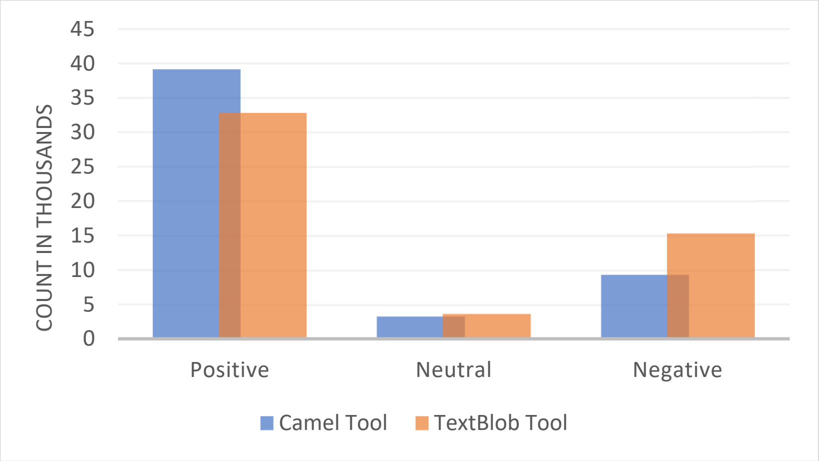

Figure 4. Comparison between sentiment ratio obtained by using the Camel tool and TextBlob.

3.3.2. Bing Liu Lexicon

Reviews in the Google Play Store have a slightly different style of language. The users often tend to write quite long sentences without caring for grammatical rules. In addition, they judge many apps' features at the same time. In this work, the Bing Liu lexicon was used. As the Bing Liu lexicon does not support Arabic, the reviews were first translated into English using Google translator APIs. The original Bing Lui lexicon contains about 6800 words extracted from product reviews, where 2006 words were labeled as positive words and the remaining 4783 words were negative words [38]. The lexicon is available at (https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html) (accessed on 20 March 2022). In this study, the lexicon was used as follows: the feature vectors were formed by adding up the frequencies of positive and negative words in each retrieved review.

3.3.3. AFINN Lexicon

The AFINN lexicon is a general-purpose lexicon that rates 2477 words for valence with an integer between −5, the most negative, and +5, the most positive review [39]. The lexicon is available publicly at (http://www2.imm.dtu.dk/pubdb/pubs/6010-full.html) (accessed on 20 March 2022). As with the Bing Lui lexicon, each review was translated into English, then the positive and negative scores were calculated by aggregating the word associations provided by the AFINN lexicon. Table 7 shows a sample of reviews from the dataset and the computed scores obtained by the AFINN lexicon.

Table 7. A sample of reviews with AFINN lexicon scores and their polarity.

Sr.	Sample Review	Translated Review	AFINN pos_score	AFINN neg_score	AFINN Polarity	Original Polarity
1	.لا يعمل ولا يستجيب أثناء التسجيل بالإضافة، كود التفعيل لا يُرسل	It does not work and does not respond at the registration stage. In addition, the verification code is not sent.	0.0	0.0	Neutral	Negative
2	.للأسف، تطبيق سيئ جدا منذ يومين أحاول التسجيل ولا يتم إرسال كود التفعيل	Unfortunately, a very bad application for two days in the tried to login and does not send the verification code?	0.0	−3.0	Negative	Negative
3	يستحق 5نجوم وأكثر	It worth 5 stars and more.	2.0	0.0	Positive	Positive
4	!لقد سجلت في التطبيق، لكن ماذا يحدث .رمز المرور ورقم الهاتف صحيح .عندما أحاول الدخول يطلب مني ادخال رقم الهوية	I have a registered in the app, but what does open? the mobile number and password are right, and it gives me wrong. And when I try entering again, it asks me to provide the identity number.	2.0	−1.0	Positive	Neutral

3.3.4. MPQA Subjectivity Lexicon

The MPQA lexicon counts the number of positive and negative words by automatically distinguishing their prior and contextual polarity [40]. The MPQA stands for Multi-Perspective Question Answering. It includes 2533 positive words and 5097 negative words (http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/) (accessed on 2 March 2022). In addition, each word is provided with a polarity annotation score and POS-tagging. Table 8 shows the MPQA polarity score for the samples presented earlier in Table 7.

Table 8. A sample of reviews with MPQA subjectivity lexicon scores and their polarity.

Sr.	MPQA pos_count	MPQA neg_count	MPQA Polarity	Original Polarity
1	0.0	0.0	Neutral	Negative
2	0.0	2.0	Negative	Negative
3	2.0	0.0	Positive	Positive
4	1.0	0.0	Positive	Neutral

3.4. Feature Extraction Techniques

Studies show that the efficacy of the ML models can be uplifted when the right feature extraction technique is applied [33,41,42]. The current study incorporates the following feature extraction techniques:

- Bag of Words (BoW)
- Term Frequency-Inverse Document Frequency (TF-IDF)
- Word2Vec Embeddings

- Concatenated Features, where the final set composes the features extracted by other feature extraction techniques. The concatenated features sets, in this work, are formed as follows:

$$Con^1_{feature} = BoW \cup \ TF_{IDF} \tag{1}$$

$$Con^2_{feature} = BoW \cup \ Word2Vec \tag{2}$$

$$Con^3_{feature} = TF_{IDF} \cup \ Word2Vec \tag{3}$$

$$Con^4_{feature} = BoW \cup \ TF_{IDF} \cup \ Word2Vec \tag{4}$$

3.5. Machine Learning Models

As stated earlier, five ML classifiers were implemented: random forest (RF), bagging, support vector machine (SVM), logistic regression (LR), and naïve Bayes (NB). The effectiveness of the proposed feature extraction techniques was also measured by conducting different experiments. For tuning hyperparameters of the used ML classifiers, the grid search algorithm with 10-fold cross-validation was used. Afterward, the hyperparameter values that yielded the highest performance measure were chosen as the final hyperparameters for each classifier, as shown in Table 9.

Table 9. The optimized hyperparameters' settings.

ML Classifier	Optimized Hyperparameters' Settings
RF	Criterion = "entropy"; max_depth = 1500; min_samples_leaf = 7; min_samples_split = 5; n_estimators = 200
Bagging	n_estimators: 100; max_samples = 0.5, max_features = 0.5
SVM	C = 0.1; Gamma = 1×10^{-4}; Kernel = "Linear"
LR	C = 1×10^{-3}, fit_intercept = True
NB	alpha = 1×10^{-5}; fit_prior = True

3.6. Evaluation Metrics

The performance of each classifier was measured by computing the following performance measures: classification accuracy, precision, recall, and F1-score. These measures are commonly used to evaluate the performance of ML models in many research areas, such as rumor detection systems [43,44], clickbait detection [33], as well as in SA [45–47].

4. Results and Analysis

This section provides a detailed description of the experimental results as well as an analysis of the results. The experiment set included many ML models, in which we evaluated their performance with the feature extraction techniques using the imbalanced dataset. First, we present the findings along with the original sentiments, as well as sentiments extracted by Arab TextBlob. Next, the dataset is enriched by adding the features extracted by Bing Lui, AFINN, and MPQA lexicons. Later, the dataset is balanced using the SMOTE technique and the performance of the ML-based models is investigated again.

4.1. Original Sentiment of the Arb-AppsReview Dataset

The experimental results in Table 10 show the performance of ML models that are trained on features extracted by BoW, keeping the sentiments of the original Arb-AppsReview dataset. According to the evaluation metrics, the NB classifier performed better than the other models. It yielded the highest accuracy of 74.25%, a precision of 0.74%, a recall of 0.743, and an F1-score of 0.735. On the contrary, LR acquired the lowest values in terms of accuracy as well as recall, whereas, in terms of precision and F1-score, RF had the worst value. In addition, from the results shown in Figure 5, LR predicted a neutral class with the highest true-positive rate, whereas, in terms of positive class and negative class, NB remains the leading ML model.

Table 10. Experiments with ML models and BoW on original sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	68.86%	0.674	0.689	0.669
Bagging	72.43%	0.732	0.719	0.725
SVM	71.26%	0.706	0.713	0.696
LR	65.87%	0.720	0.659	0.675
NB	74.25%	0.740	0.743	0.735

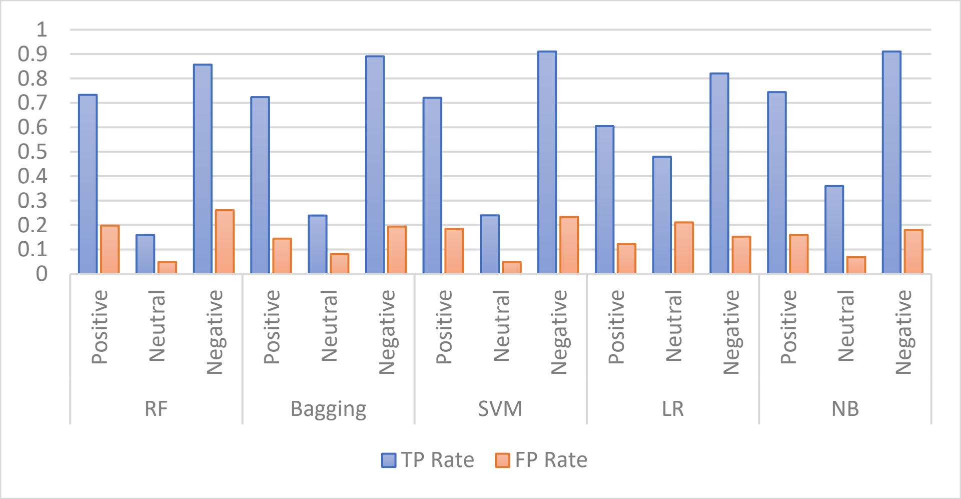

Figure 5. True-positive rate (TP Rate) and false-positive rate (FP Rate) of ML models with original sentiments using BoW.

Similarly, with features extracted by TF-IDF, NB produced the highest accuracy of 68.86%, as well as the highest precision, recall, and F1-score of 0.690, 0.689, and 0.671, respectively, as shown in Table 11. Meanwhile, the lowest accuracy was acquired by RF of 58.67% and LR of 58.68%.

Table 11. Experiments with ML models and TF-IDF on original sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	58.67%	0.718	0.587	0.634
Bagging	67.07%	0.640	0.671	0.609
SVM	61.68%	0.708	0.617	0.611
LR	58.68%	0.733	0.587	0.610
NB	68.86%	0.690	0.689	0.671

Although the NB model achieved the highest accuracy, it did not deliver optimized results in the prediction of the positive class, as the TP rate achieved by the bagging algorithm was the highest as compared to other ML models, as shown in Figure 6.

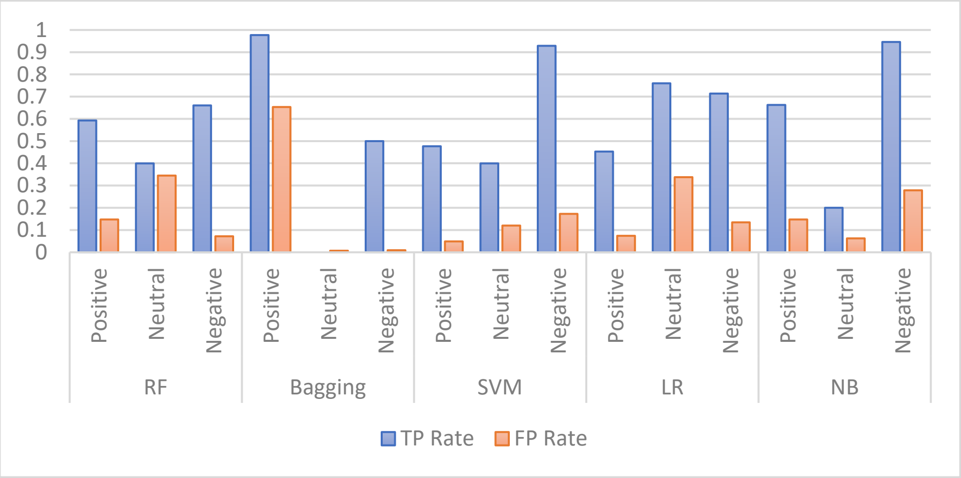

Figure 6. True-positive rate (TP Rate) and false-positive rate (FP Rate) of ML models with original sentiments using TF-IDF.

In the case of the concatenated feature as well as the Word2Vec representation, as shown in Tables 12 and 13, RF benefited more, and the highest accuracy achieved was when the features were extracted by Word2Vec. Additionally, the performance of LR improved with the features extracted by joining the BoW and TF-IDF approaches, $Con^1_{feature}$. In addition, using the concatenated features, $Con^4_{feature}$ led to a notable improvement in the performance of SVM. Similarly, the performance of LR increased with combining features using $Con^2_{feature}$.

Table 12. Experiments with ML models and concatenated features on original sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	73.6527%	0.741	0.737	0.701
Bagging	69.4611%	0.671	0.695	0.649
SVM	75.4491%	0.742	0.754	0.734
LR	73.0539%	0.721	0.731	0.723
NB	73.0539%	0.725	0.731	0.723

Table 13. Experiments with ML models and Word2Vec on original sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	76.6467%	0.809	0.766	0.726
Bagging	74.2515%	0.693	0.743	0.697
SVM	70.6587%	0.529	0.707	0.605
LR	67.6647%	0.725	0.677	0.695
NB	68.8623%	0.663	0.689	0.670

In addition, from Figures 7 and 8, it can be observed that most ML classifiers performed well in predicting the positive class. In terms of the TP rate, SVM yielded the highest score for predicting the positive class, whereas, in the case of the negative class, NB outperformed the other models.

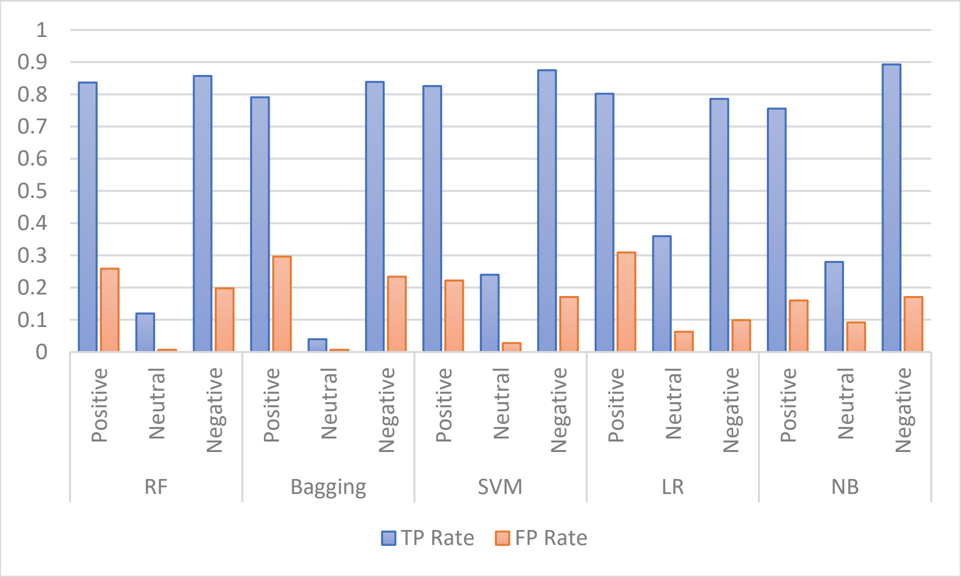

Figure 7. True-positive rate (TP Rate) and false-positive rate (FP Rate) of ML models with original sentiments using concatenated features.

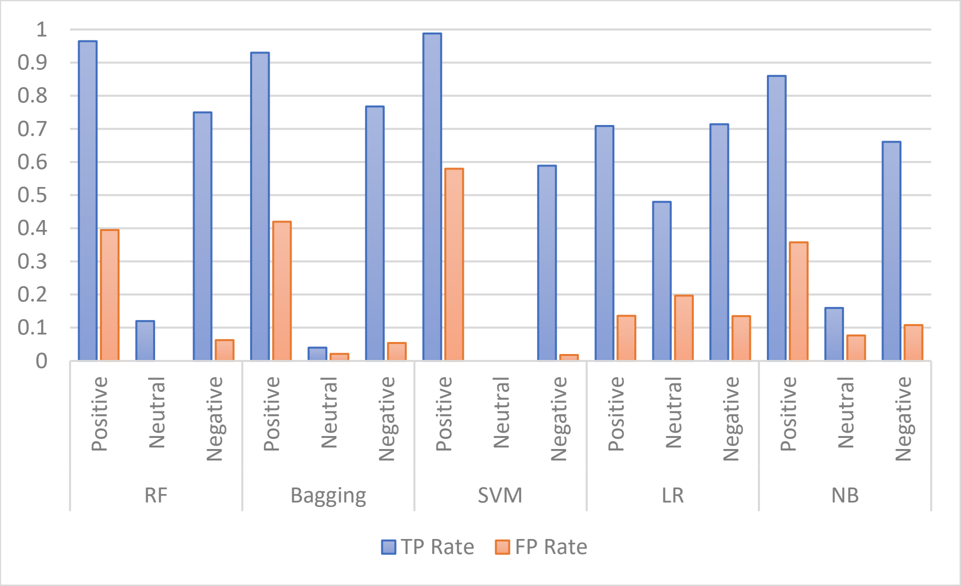

Figure 8. True-positive rate (TP Rate) and false-positive rate (FP Rate) of ML models with original sentiments using Word2Vec.

4.2. TextBlob Sentiment of the Arb-AppsReview Dataset

As stated earlier, the original Arb-AppsReview dataset was labeled using the Camel tool. After applying the TextBlob tool for Arabic (TextBlob-ar 0.0.2 extension), the distribution of polarity of sentiment changed (see Figure 4). Based on the results presented in Tables 14–17, the following findings were observed, and can be summarized as follows:

- All classifiers benefited more when the extracted features were concatenated. Among all the classifiers, SVM yielded the highest accuracy of 91.67%, with 0.913 precision, 0.927 recall, and 0.920 F1-score.
- Despite the feature extraction techniques used, all classifiers performed well when the TextBlob tool was used, which proves the efficacy of using TextBlob sentiments.
- In terms of F1-score, the SVM classifier outperformed all other ML models. It provided the highest F1-score of 0.920, followed by the bagging method.

- In all experiments conducted using Word2Vec, the performance of ML models was better compared to the results obtained when BoW or TF-IDF techniques were used.

Table 14. Experiments with ML models and BoW on Arab TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	86.43%	0.860	0.796	0.827
Bagging	84.56%	0.851	0.903	0.876
SVM	89.65%	0.898	0.898	0.898
LR	85.78%	0.856	0.857	0.856
NB	87.21%	0.873	0.887	0.880

Table 15. Experiments with ML models and TF-IDF on Arab TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	85.64%	0.793	0.876	0.832
Bagging	86.26%	0.872	0.893	0.882
SVM	84.34%	0.798	0.968	0.875
LR	82.72%	0.870	0.827	0.848
NB	89.52%	0.886	0.873	0.879

Table 16. Experiments with ML models and Word2Vec on Arab TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	88.43%	0.863	0.921	0.891
Bagging	87.86%	0.880	0.942	0.910
SVM	89.77%	0.891	0.908	0.899
LR	89.82%	0.891	0.912	0.901
NB	88.42%	0.870	0.832	0.851

Table 17. Experiments with ML models and concatenated features on Arab TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	90.37%	0.902	0.910	0.907
Bagging	90.08%	0.894	0.900	0.897
SVM	91.67%	0.913	0.927	0.920
LR	90.68%	0.900	0.920	0.900
NB	89.81%	0.876	0.881	0.878

4.3. Enriched Dataset with Features Extracted by the Lexicons

The efficacy of using TextBlob sentiments was obviously proven, as presented in Section 4.2. The current subsection provides an overview of the performance of the ML models in terms of using features extracted from the lexicons. For this purpose, only the features obtained by the Bing Lui lexicon, AFINN lexicon, MPQA lexicon, and their different combinations were used.

4.3.1. Comparative Analysis of ML Models with TextBlob Sentiments Using Bing Lui Lexicon, AFINN Lexicon, and MPQA Lexicon

Tables 18–20 show how the ML models performed with the lexicons used. Among all the investigated lexicons, the AFINN yielded the best results. It can be observed that, in terms of F1-score, the bagging classifier benefited more when the AFFIN lexicon was used as a feature extraction technique, followed by LR. The highest F1-score achieved was 0.758, which is better than the score obtained by the BoW, TF-IDF, concatenated features, and Word2Vec techniques with the original Camel-based dataset. For this reason, the next subsection presents the impact of concatenating several lexicon-based features with the TextBlob sentiment.

Table 18. Experiments with ML models and the Bing Lui lexicon on the TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	64.67%	0.565	0.647	0.601
Bagging	65.87%	0.667	0.659	0.706
SVM	66.47%	0.670	0.665	0.713
LR	66.48%	0.670	0.665	0.475
NB	66.48%	0.644	0.665	0.619

Table 19. Experiments with ML models and the AFFIN lexicon on the TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	68.26%	0.612	0.683	0.628
Bagging	71.86%	0.764	0.719	0.758
SVM	70.06%	0.649	0.701	0.644
LR	69.46%	0.598	0.695	0.638
NB	68.86%	0.659	0.689	0.658

Table 20. Experiments with ML models and the MPQA lexicon on the TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	67.67%	0.599	0.677	0.628
Bagging	69.46%	0.711	0.695	0.736
SVM	67.67%	0.596	0.677	0.624
LR	68.86%	0.695	0.689	0.731
NB	68.86%	0.648	0.689	0.651

4.3.2. Comparative Analysis of ML Models with TextBlob Sentiments Using Different Combinations of Lexicons

It was noticed that the performance of ML classifiers was improved in terms of F1-score when the lexicons were used as a feature extraction technique. This section investigates the performance of the ML models, combining several lexicons on the TextBlob sentiment. First, we have investigated the impact of lexicons without concatenating the extracted features by BoW, TF-IDF, Word2Vec, and the concatenated features. In the next section, the feature extraction techniques are combined with a lexicon-based model and formulate one dataset. As shown in Tables 21–24, the following findings are noted:

- RF and SVM improved in terms of accuracy, with 70.08% and 71.26%, respectively.
- Using Bing Lui + MPQA lexicons led to degrading the performance of NB.
- A combination of the Bing Lui lexicon and the AFINN lexicon, in most cases, yielded the best results.
- The combination of all extracted features does not always improve the performance of the ML classifiers. This means that the researchers must carefully select the lexicon and investigate the performance of each classifier using all the possible combinations.

Table 21. Experiments with ML models and Bing Lui + AFINN lexicons on the TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	70.06%	0.638	0.701	0.650
Bagging	71.86%	0.764	0.719	0.758
SVM	68.86%	0.616	0.689	0.631
LR	67.67%	0.620	0.677	0.636
NB	65.27%	0.615	0.653	0.619

Table 22. Experiments with ML models and Bing Lui + MPQA lexicons on the TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	68.26%	0.629	0.683	0.646
Bagging	71.86%	0.751	0.719	0.712
SVM	71.26%	0.747	0.713	0.572
LR	64.67%	0.563	0.647	0.602
NB	59.88%	0.600	0.599	0.575

Table 23. Experiments with ML models and AFINN + MPQA lexicons on the TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	65.87%	0.588	0.659	0.612
Bagging	70.06%	0.623	0.701	0.639
SVM	68.86%	0.601	0.689	0.631
LR	68.26%	0.593	0.683	0.631
NB	68.86%	0.643	0.689	0.656

Table 24. Experiments with ML models and all lexicons on the TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	69.46%	0.647	0.695	0.654
Bagging	71.26%	0.632	0.713	0.649
SVM	69.46%	0.636	0.695	0.638
LR	65.87%	0.579	0.659	0.613
NB	64.67%	0.617	0.647	0.620

4.4. Arab TextBlob Sentiment of the Arb-AppsReview Dataset with Enriched Features Extracted by Lexicons

Tables 25–28 show that the highest accuracy score of 93.17% was achieved by SVM, with the enriched features extracted by lexicons. The results confirm the efficiency of SVM when it is integrated with the concatenated features and the features extracted by the proposed lexicons. In addition, the results show that using TextBlob for revising the sentiment of the original dataset is an efficient tool that leads to improving the performance of the ML models. It is also obvious that the performance of ML models is affected differently by the word representation approaches, e.g., RF yielded an accuracy score of 86.14% when the Word2Vec technique was used, whilst it yielded accuracy scores of 85.44% and 84.44% when BoW and TF-IDF were used, respectively. Similarly, the bagging approach yielded different accuracy scores where the BoW technique afforded the highest score. Figure 9 shows a comparison between the results presented in Table 17 and the results obtained by concatenating all features and lexicons with TextBlob sentiment. The results show that the performance of the ML classifiers benefited more when TextBlob sentiment and the features were extracted by all lexicons and word representation techniques.

Table 25. Experiments with ML models and BoW and all lexicons on Arab TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	85.44%	0.867	0.854	0.823
Bagging	88.43%	0.813	0.884	0.848
SVM	84.25%	0.834	0.843	0.830
LR	82.46%	0.834	0.825	0.828
NB	84.85%	0.835	0.849	0.837

Table 26. Experiments with ML models and TF-IDF and all lexicons on Arab TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	84.25%	0.887	0.843	0.812
Bagging	80.06%	0.861	0.725	0.786
SVM	76.47%	0.818	0.765	0.752
LR	83.13%	0.857	0.831	0.827
NB	84.25%	0.832	0.843	0.832

Table 27. Experiments with ML models and Word2Vec and all lexicons on Arab TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	86.14%	0.848	0.860	0.823
Bagging	84.25%	0.788	0.843	0.799
SVM	80.66%	0.744	0.807	0.752
LR	86.10%	0.880	0.860	0.867
NB	78.86%	0.763	0.789	0.770

Table 28. Experiments with ML models and concatenated features and all lexicons on Arab TextBlob sentiments.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	92.25%	0.921	0.928	0.924
Bagging	90.78%	0.914	0.931	0.922
SVM	93.17%	0.939	0.927	0.933
LR	91.27%	0.908	0.913	0.910
NB	90.18%	0.897	0.893	0.895

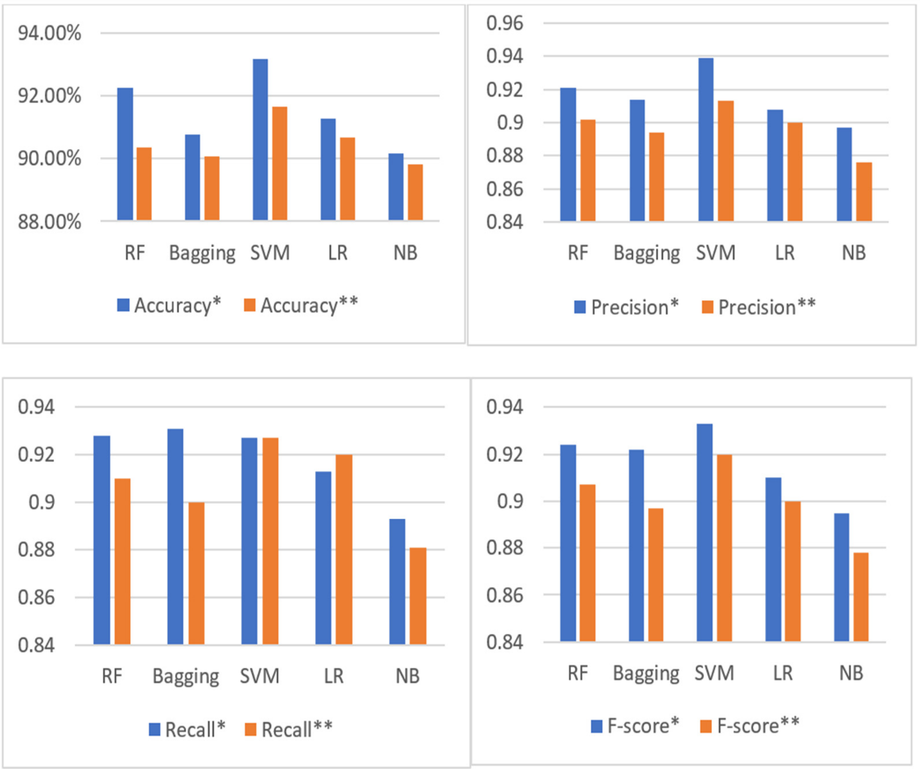

Figure 9. Performance of ML models with TextBlob sentiment and the concatenated features + lexicons: * denotes that TextBlob sentiment is used with the concatenated features extracted by the word representation techniques; ** TextBlob sentiment is used with concatenated features + lexicons.

4.5. Comparative Analysis of ML Models with TextBlob Sentiments Using Balanced Dataset

As shown in the previous sections, the performance of ML models was demonstrated using an imbalanced dataset. In this section, the experiment was performed using a balanced dataset. There are two main techniques that are widely used for solving imbalanced dataset problems: (i) random under-sampling, where the size of the dataset is reduced by removing some samples of the majority class, and (ii) the oversampling technique, in which the number of samples of the minority class is increased [48]. In this work, the synthetic minority oversampling technique (SMOTE) was used for oversampling [49]. Table 29 shows the number of samples after re-sampling was applied.

Table 29. Number of samples after applying re-sampling.

Category	Original Dataset Size	Under-Sampling	Oversampling
Positive Reviews	39,165	9304	39,165
Negative Reviews	9304	9304	39,165
Total	48,469	18,608	78,330

As shown in Table 29, the number of samples per class changed according to the applied strategy. After re-sampling, the data were split into training and testing sets using 10-fold cross-validation. The previous sections of this paper showed the performance of the proposed approach using the original dataset without re-sampling, and the best-achieved results were obtained when the concatenated features and all lexicons on Arab TextBlob sentiments were used. Hence, this section shows only the results of the experiments with under-sampling, and experiments with oversampling using the concatenated features and all lexicons on Arab TextBlob sentiments.

4.5.1. Performance of Models on Balanced Dataset Using Under-Sampling

As shown earlier, the proposed approach achieved the best performance when the concatenated features and all lexicons on Arab TextBlob sentiments were used. To investigate the robustness of the model and to avoid any overfitting, further experiments were performed using a balanced dataset with a random under-sampling technique. The obtained results of the under-sampled data are shown in Table 30.

Table 30. Performance results of ML models using the under-sampling dataset and the concatenated features and all lexicons.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	87.14%	0.872	0.873	0.872
Bagging	88.36%	0.881	0.883	0.882
SVM	87.39%	0.873	0.873	0.873
LR	84.92%	0.849	0.851	0.850
NB	82.11%	0.821	0.821	0.821

The results show that the performance of the selected models has been degraded. Among all the ML classifiers, the NB classifier had the worst accuracy, of 82.11%. The reason behind this degradation is the reduction in the size of the dataset. In contrast to the findings shown in Table 28, the bagging classifier had the highest accuracy and F1-score, with values of 88.36% and 0.882, respectively. In addition, the performance of SVM was negatively affected.

4.5.2. Performance of Models on Balanced Dataset Using SMOTE Technique

Table 31 shows the performance of the ML models with respect to the SMOTE balanced dataset. The results show that the performance of machine learning models improved significantly when the SMOTE technique was used. In addition, SVM outperformed all other models in terms of all evaluation metrics.

Table 31. Performance results of ML models using the SMOTE dataset and the concatenated features and all lexicons.

Classifier	Accuracy	Precision	Recall	F1-Score
RF	93.87%	0.935	0.936	0.935
Bagging	91.89%	0.919	0.919	0.919
SVM	94.38%	0.943	0.943	0.943
LR	92.68%	0.926	0.928	0.927
NB	93.11%	0.930	0.931	0.930

By comparing the performance of all ML models before and after balancing the dataset, it is obvious that the NB classifier benefited more when the SMOTE technique was applied. On the other hand, the NB classifier was extremely affected, as shown in Figure 10.

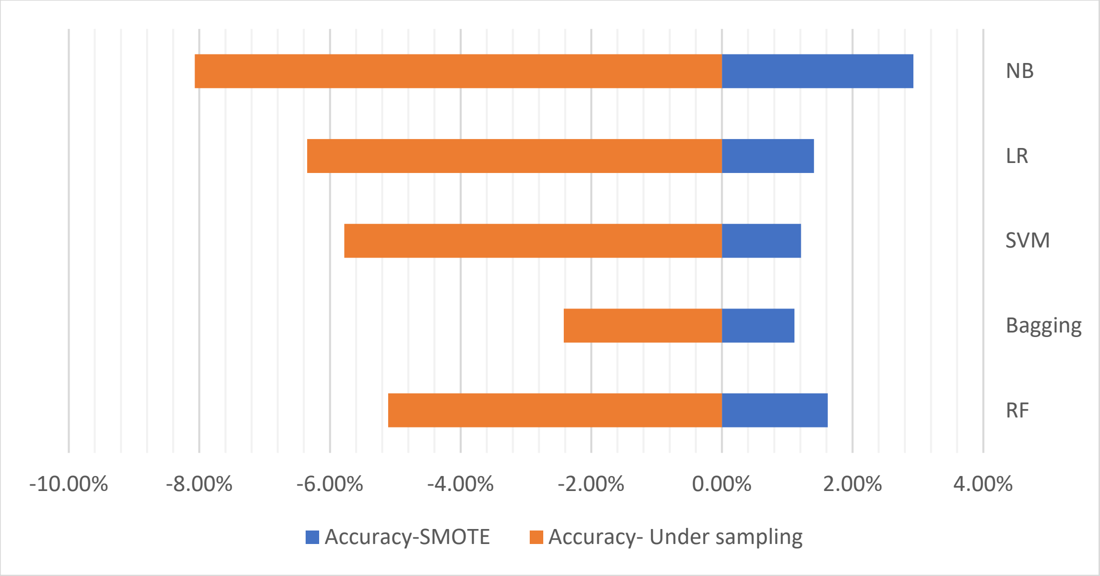

Figure 10. Percentage of change of the ML classifiers with respect to the balancing techniques.

In addition, the outcomes of the proposed methods (ML models using SMOTE dataset and the concatenated features and all lexicons) were compared with a recent and similar study [47] that used four ML methods for Arabic sentiment analysis of users' opinions of governmental mobile applications. The study [47] was one of the earliest to explore users' opinions about governmental apps in Saudi Arabia. They applied DTree, SVM, KNN, and NB classifiers on a new Arabic dataset that included 7759 reviews collected from Google Play and the App Store. The results of [47] showed that KNN outperformed the other methods with the accuracy of 78.46%. Table 32 shows the comparison results of the best-performing methods in this study and [47]. It is obviously shown that the proposed method obtained superior performance to the ML methods used in the previous recent study [39] using all evaluation criteria.

Table 32. Comparison results of the proposed ML models with a previous study.

Classifier	Accuracy	Precision	Recall	F1-Score
The proposed method (SVM)	94.38%	0.943	0.943	0.943
The previous study [49] (KNN)	78.46%	0.799	0.780	0.790

5. Conclusions

This paper presented the Arabic sentiment analysis on the reviews of six governmental mobile applications from Google Play and the App Store. Several ML models were applied to the AppsReview dataset, which includes 51k reviews. These ML methods were: RF, bagging, SVM, LR, and NB. In the conducted experiments, we evaluated the ML performance by integrating different feature extraction techniques, which were Bing Lui, AFINN, and MPQA lexicons. In addition, the performance of the ML models was investigated using an imbalanced and a balanced dataset. As balancing techniques, both under-sampling and oversampling were used. In this regard, the SMOTE technique was applied as an oversampling technique. The experimental results showed that when the features were extracted by BoW using the sentiments of the original Arb-AppsReview dataset, the NB classifier performed better than the other models (74.25% for accuracy). Then, when the TextBlob sentiment tool was applied to the Arb-AppsReview dataset, it was found that all classifiers performed better when the extracted features were concatenated. Among all the classifiers, SVM yielded the highest accuracy of 91.67%. Then, the findings showed that the highest accuracy score of 93.17% was achieved by SVM when it was integrated with the concatenated features and the features extracted by the proposed lexicons. Finally, the outcomes of the ML models using the SMOTE technique, the concatenated features, and all lexicons obtained the best results. For instance, SVM with these preprocessing methods obtained an accuracy of 94.38%, which overcame all other models. This study recommends applying the used feature engineering methods with the SMOTE oversampling technique to obtain better ASA results. Since the proposed model relies on the translation of reviews using google translation, the quality of the model might be affected by the quality of the translation. For future work, it is recommended to investigate the machine translation techniques to obtain high-quality translations that might increase the model performance. Additionally, it is recommended to explore the effect of applying different feature extraction and selection methods to the dataset used in this research and conduct more experiments that apply different combinations of deep learning methods on the Arb-AppsReview dataset to enhance the Arabic sentiment analysis.

Author Contributions: Conceptualization, M.A.-S., F.S. and M.H.; methodology, M.A.-S., F.S. and M.A.A.-H.; software, M.A.-S.; validation, M.H. and M.A.A.-H.; formal analysis, M.A.-S., F.S. and M.H.; investigation, M.H. and M.A.A.-H.; resources, M.H., M.A.-S. and M.A.A.-H.; data curation, M.A.-S.; writing—original draft preparation, M.A.-S., F.S. and M.H.; writing—review and editing, M.A.-S., F.S. and M.H.; visualization, M.A.-S.; supervision, M.A.-S. and F.S.; project administration, M.H. and M.A.A.-H.; funding acquisition, M.H. and M.A.A.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the Deanship of Scientific Research at the University of Qassim, Saudi Arabia, through project number 10278-coc-2020-1-3-I.

Data Availability Statement: The dataset can be obtained from: https://github.com/Arb-AppsReview/Arb-AppsReview/blob/main/apps_ar_reviews_dataset.csv (accessed on 11 November 2021).

Acknowledgments: The authors gratefully acknowledge Qassim University, represented by the Deanship of Scientific Research, for the financial support for this research under the number 10278-coc-2020-1-3-I during the academic year 1441 AH/2020 AD.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xia, R.; Zong, C.; Li, S. Ensemble of feature sets and classification algorithms for sentiment classification. *Inf. Sci.* **2011**, *181*, 1138–1152. [CrossRef]
2. Alsaeedi, A.; Khan, M.Z. A study on sentiment analysis techniques of Twitter data. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 361–374. [CrossRef]
3. Alomari, K.M.; ElSherif, H.M.; Shaalan, K. Arabic tweets sentimental analysis using machine learning. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*; Springer: Cham, Switzerland, 2017; pp. 602–610.
4. Abuelenin, S.; Elmougy, S.; Naguib, E. Twitter sentiment analysis for arabic tweets. In *International Conference on Advanced Intelligent Systems and Informatics*; Springer: Cham, Switzerland, 2017; pp. 467–476.
5. Shoukry, A.; Rafea, A. Sentence-level Arabic sentiment analysis. In Proceedings of the 2012 International Conference on Collaboration Technologies and Systems (CTS), Denver, CO, USA, 21–25 May 2012; pp. 546–550.

6. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 606–615.
7. Abdullah, M.; Hadzikadicy, M.; Shaikhz, S. SEDAT: Sentiment and emotion detection in Arabic text using CNN-LSTM deep learning. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 835–840.
8. Boudad, N.; Faizi, R.; Thami, R.O.H.; Chiheb, R. Sentiment analysis in Arabic: A review of the literature. *Ain Shams Eng. J.* **2018**, *9*, 2479–2490. [CrossRef]
9. Rushdi-Saleh, M.; Martín-Valdivia, M.T.; Ureña-López, L.A.; Perea-Ortega, J.M. OCA: Opinion corpus for Arabic. *J. Am. Soc. Inf. Sci. Technol.* **2011**, *62*, 2045–2054. [CrossRef]
10. Ghallab, A.; Mohsen, A.; Ali, Y. Arabic Sentiment Analysis: A Systematic Literature Review. *Appl. Comput. Intell. Soft Comput.* **2020**, *2020*, 7403128. [CrossRef]
11. Tsarfaty, R.; Seddah, D.; Goldberg, Y.; Kübler, S.; Versley, Y.; Candito, M.; Tounsi, L. Statistical parsing of morphologically rich languages (spmrl) what, how and whither. In Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, Los Angeles, CA, USA, 5 June 2010; pp. 1–12.
12. Elouardighi, A.; Maghfour, M.; Hammia, H.; Aazi, F.-Z. A machine Learning approach for sentiment analysis in the standard or dialectal Arabic Facebook comments. In Proceedings of the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, Morocco, 24–26 October 2017; pp. 1–8. [CrossRef]
13. Hammad, A.; El-Halees, A. An approach for detecting spam in Arabic opinion reviews. *Int. Arab. J. Inf. Technol.* **2013**, *12*, 1–8.
14. Brahimi, B.; Touahria, M.; Tari, A. Data and Text Mining Techniques for Classifying Arabic Tweet Polarity. *J. Digit. Inf. Manag.* **2016**, *14*, 1.
15. Ihnaini, B.; Mahmuddin, M. Lexicon-based sentiment analysis of arabic tweets: A survey. *J. Eng. Appl. Sci.* **2018**, *13*, 7313–7322.
16. 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]
17. Alotaibi, S.; Mehmood, R.; Katib, I. Sentiment analysis of arabic tweets in smart cities: A review of saudi dia-lect. In Proceedings of the 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), Rome, Italy, 10–13 June 2019; pp. 330–335.
18. Mustafa, H.H.; Mohamed, A.; Elzanfaly, D.S. An enhanced approach for arabic sentiment analysis. *Int. J. Artif. Intell. Appl. (IJAIA)* **2017**, *8*, 5. [CrossRef]
19. Gamal, D.; Alfonse, M.; El-Horbaty, E.S.M.; Salem, A.B.M. Implementation of machine learning algorithms in Ara-bic sentiment analysis using N-gram features. *Procedia Comput. Sci.* **2019**, *154*, 332–340. [CrossRef]
20. Touahri, I.; Mazroui, A. Studying the effect of characteristic vector alteration on Arabic sentiment classification. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *33*, 890–898. [CrossRef]
21. Aloqaily, A.; Al-Hassan, M.; Salah, K.; Elshqeirat, B.; Almashagbah, M. Sentiment analysis for arabic tweets da-tasets: Lexicon-based and machine learning approaches. *J. Theor. Appl. Inf. Technol.* **2020**, *98*, 4.
22. Aljameel, S.S.; Alabbad, D.A.; Alzahrani, N.A.; AlQarni, S.M.; AlAmoudi, F.A.; Babili, L.M.; Aljaafary, S.K.; Alshamrani, F.M. A Sentiment Analysis Approach to Predict an Individual's Awareness of the Precautionary Procedures to Prevent COVID-19 Outbreaks in Saudi Arabia. *Int. J. Environ. Res. Public Health* **2020**, *18*, 218. [CrossRef] [PubMed]
23. Althagafi, A.; Althobaiti, G.; Alhakami, H.; Alsubait, T. Arabic Tweets Sentiment Analysis about Online Learning during COVID-19 in Saudi Arabia. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 620–625. [CrossRef]
24. Alassaf, M.; Qamar, A.M. Improving sentiment analysis of Arabic tweets by One-Way ANOVA. *J. King Saud Univ. Comput. Inf. Sci.* 2020, *in press*. [CrossRef]
25. Heikal, M.; Torki, M.; El-Makky, N. Sentiment Analysis of Arabic Tweets using Deep Learning. *Procedia Comput. Sci.* **2018**, *142*, 114–122. [CrossRef]
26. Al-Twairesh, N.; Al-Negheimish, H. Surface and Deep Features Ensemble for Sentiment Analysis of Arabic Tweets. *IEEE Access* **2019**, *7*, 84122–84131. [CrossRef]
27. Mohammed, A.; Kora, R. Deep learning approaches for Arabic sentiment analysis. *Soc. Netw. Anal. Min.* **2019**, *9*, 52. [CrossRef]
28. Khalil, E.A.H.; El Houby, E.M.F.; Mohamed, H.K. Deep learning for emotion analysis in Arabic tweets. *J. Big Data* **2021**, *8*, 1–15. [CrossRef]
29. Alharbi, N.H.; Alkhateeb, J.H. Sentiment Analysis of Arabic Tweets Related to COVID-19 Using Deep Neural Network. In Proceedings of the 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Virtual Conference, 4–5 July 2021; pp. 1–11.
30. Shahi, A.M.; Issac, B.; Modapothala, J.R. Intelligent Corporate Sustainability report scoring solution using machine learning approach to text categorization. In Proceedings of the 2012 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT), Kuala Lumpur, Malaysia, 6–9 October 2012; pp. 227–232. [CrossRef]
31. Umer, M.; Ashraf, I.; Mehmood, A.; Kumari, S.; Ullah, S.; Sang Choi, G. Sentiment analysis of tweets using a uni-fied convolutional neural network-long short-term memory network model. *Comput. Intell.* **2021**, *37*, 409–434. [CrossRef]
32. Al-Sarem, M.; Saeed, F.; Al-Mekhlafi, Z.G.; Mohammed, B.A.; Hadwan, M.; Al-Hadhrami, T.; Alshammari, M.T.; Alreshidi, A.; Alshammari, T.S. An Improved Multiple Features and Machine Learning-Based Approach for Detecting Clickbait News on Social Networks. *Appl. Sci.* **2021**, *11*, 9487. [CrossRef]

33. Al-Sarem, M.; Al-Harby, M.; Saeed, F.; Hezzam, E.A. Machine Learning Classifiers with Preprocessing Techniques for Rumor Detection on Social Media: An Empirical Study. *Int. J. Cloud Computing.* 2021, *in press*.
34. Al-Sarem, M.; Saeed, F.; Alsaeedi, A.; Boulila, W.; Al-Hadhrami, T. Ensemble Methods for Instance-Based Arabic Language Authorship Attribution. *IEEE Access* **2020**, *8*, 17331–17345. [CrossRef]
35. Rustam, F.; Ashraf, I.; Mehmood, A.; Ullah, S.; Choi, G.S. Tweets Classification on the Base of Sentiments for US Airline Companies. *Entropy* **2019**, *21*, 1078. [CrossRef]
36. Gaye, B.; Zhang, D.; Wulamu, A. A Tweet Sentiment Classification Approach Using a Hybrid Stacked Ensemble Technique. *Information* **2021**, *12*, 374. [CrossRef]
37. Loria, S. Textblob Documentation. *Release 0.15* **2018**, *2*, 269.
38. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 168–177.
39. Nielsen, F.Å. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv* **2011**, arXiv:1103.2903.
40. Wilson, T.; Wiebe, J.; Hoffmann, P. Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; pp. 347–354.
41. Heaton, J. An empirical analysis of feature engineering for predictive modeling. In Proceedings of the IEEE SoutheastCon 2016, Norfolk, VA, USA, 30 March–3 April 2016; pp. 1–6. [CrossRef]
42. Al-Sarem, M.; Saeed, F.; Boulila, W.; Emara, A.H.; Al-Mohaimeed, M.; Errais, M. Feature Selection and Classification Using CatBoost Method for Improving the Performance of Predicting Parkinson's Disease. In *Advances on Smart and Soft Computing*; Springer: Singapore, 2020; pp. 189–199. [CrossRef]
43. Al-Sarem, M.; Alsaeedi, A.; Saeed, F.; Boulila, W.; AmeerBakhsh, O. A Novel Hybrid Deep Learning Model for De-tecting COVID-19-Related Rumors on Social Media Based on LSTM and Concatenated Parallel CNNs. *Appl. Sci.* **2021**, *11*, 7940. [CrossRef]
44. Alsaeedi, A.; Al-Sarem, M. Detecting Rumors on Social Media Based on a CNN Deep Learning Technique. *Arab. J. Sci. Eng.* **2020**, *45*, 10813–10844. [CrossRef]
45. Zhao, Z.; Hao, Z.; Wang, G.; Mao, D.; Zhang, B.; Zuo, M.; Yen, J.; Tu, G. Sentiment Analysis of Review Data Using Blockchain and LSTM to Improve Regulation for a Sustainable Market. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *17*, 1–19. [CrossRef]
46. Abo, M.E.M.; Idris, N.; Mahmud, R.; Qazi, A.; Hashem, I.A.T.; Maitama, J.Z.; Yang, S. A Multi-Criteria Ap-proach for Arabic Dialect Sentiment Analysis for Online Reviews: Exploiting Optimal Machine Learning Algorithm Selection. *Sustainability* **2021**, *13*, 10018. [CrossRef]
47. Hadwan, M.; Al-Hagery, M.; Al-Sarem, M.; Saeed, F. Arabic Sentiment Analysis of Users' Opinions of Govern-mental Mobile Applications. *Comput. Mater. Contin.* **2022**, *72*, 4675–4689. [CrossRef]
48. Rupapara, V.; Rustam, F.; Shahzad, H.F.; Mehmood, A.; Ashraf, I.; Choi, G.S. Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification Using RVVC Model. *IEEE Access* **2021**, *9*, 78621–78634. [CrossRef]
49. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]