algorithms

MDPI

Article

Developing Analytical Tools for Arabic Sentiment Analysis of COVID-19 Data

Naglaa Abdelhady *, Ibrahim E. Elsemman, Mohammed F. Farghally and Taysir Hassan A. Soliman

Department of Information Systems, Faculty of Computers and Information, Assiut University, Assiut 2071515, Egypt; elsemman@aun.edu.eg (I.E.E.); mfseddik@aun.edu.eg (M.F.F.); taysirhs@aun.edu.eg (T.H.A.S.)
* Correspondence: naglaaelhady@aun.edu.eg

Abstract: Due to the widespread distribution of coronavirus and the existence of a massive quantity of data on social networking sites, particularly Twitter, there was an urgent need to develop a model that evaluates users' emotions and determines how they feel about the pandemic. However, the absence of resources to assist Sentiment Analysis (SA) in Arabic hampered the completion of this endeavor. This work presents the ArSentiCOVID lexicon, the first and largest Arabic SA lexicon for COVID-19 that handles negation and emojis. We design a lexicon-based sentiment analyzer tool that depends mainly on the ArSentiCOVID lexicon to perform a three-way classification. Furthermore, we employ the sentiment analyzer to automatically assemble 42K annotated Arabic tweets for COVID-19. We conduct two experiments. First, we test the effect of applying negation and emoji rules to the created lexicon. The results indicate that after applying the emoji, negation, and both rules, the F-score improved by 2.13%, 4.13%, and 6.13%, respectively. Second, we applied an ensemble method that combines four feature groups (n-grams, negation, polarity, and emojis) as input features for eight Machine Learning (ML) classifiers. The results reveal that Random Forest (RF) and Support Vector Machine (SVM) classifiers work best, and that the four feature groups combined are best for representing features produced the maximum accuracy of (92.21%), precision (92.23%), recall (92.21%), and F-score (92.23%) with 3.2% improvement over the base model.

Keywords: sentiment analysis; Twitter; Arabic lexicon; Arabic annotated datasets; COVID-19; negation; emoticons

Citation: Abdelhady, N.; Elsemman, I.E.; Farghally, M.F.; Soliman, T.H.A. Developing Analytical Tools for Arabic Sentiment Analysis of COVID-19 Data. *Algorithms* **2023**, *16*, 318. https://doi.org/10.3390/a16070318

Academic Editor: Frank Werner

Received: 28 May 2023
Revised: 19 June 2023
Accepted: 26 June 2023
Published: 29 June 2023

1. Introduction

Many diseases are associated with the coronavirus family. They have the potential to induce symptoms extending from the common cold to Middle East respiratory syndrome and severe acute respiratory syndrome. COVID-19 is the most novel form of coronavirus that has never been identified in human beings before. Coronaviruses are widely spread among mammals have been transferred from animals to humans. The World Health Organization (WHO) announced COVID-19 as a global pandemic where almost everyone in the world was under the high probability of exposure to COVID-19 in one way or another (https://www.who.int/emergencies/diseases/novel-coronavirus-2019, accessed on 2 December 2022). Social media plays a key role in transferring information about the novel coronavirus outbreaks. Twitter is considered one of the most powerful platforms in processing the COVID-19 information, with around 500 million tweets per day and 5787 tweets per second. Adults significantly increased Twitter usage during the pandemic (https://blog.hootsuite.com/twitter-statistics/, accessed on 22 December 2022). People use Twitter to share their personal experiences and spread good vibes among those who are infected. SA is a method for classifying the opinions expressed at word, sentence, or even document level [1]. SA has gained significant attention from researchers in recent years, and significant progress was made for several languages, most notably English. However,

as this study is confined to the Arabic language, SA is still a difficult undertaking. The difficulties encountered in Arabic SA are due to the language's inflectional structure [2], the prevalence of dialects, the absence of resources and tools for Arabic dialects, the restriction of Arabic sentiment lexicons, and the use of compound words and idioms, among other things. There are two basic methodologies reported in the literature for SA. A machine learning approach that employs annotated data as training data for building a model that can predict the classes of unseen data along with one of the ML classifiers (NB, SVM, Decision Tree (DT)). The second approach is semantic orientation, which employs sentiment lexicons and other linguistic resources to determine the sentiment of a particular phrase depend on the polarity of its terms [3].

SA has a lot of challenges, one of the most challenging to solve is negation in general terms. In SA, negation words can have an impact on the meaning of a statement, resulting in a shift in the sentiment orientation. For example, negation words (such as "not") and redirection emotion of phrases (such as "good"). in order to address this challenge, which is complex since it requires the recognition of negation words and a subsequent determination of the impacted terms, also known as negation scope, must be determined using semantic or syntactic representations [4]. Numerous studies have addressed the issue of recognizing negation phrases and the scope of negation to enhance the effectiveness of machine learning approach, such as those conducted by [5–7], among others.

Since there are not many Arabic corpora and lexicons available to the public for SA [8,9], especially for COVID-19 [10–12], the importance of this publication is in developing a lexicon and corpus for COVID-19 and making them available. An additional focus of this study is to describe how to construct an effective SA tool using a lexicon and to examine the relationship between lexicon-based rules, such as negation and emojis. Our sentiment analyzer also has the ability to deal with both negation and emojis, which are usually not used in current studies on Arabic SA, according to what we know so far. Finally, we evaluate the lexicon's performance with two different experiments. One performing lexicon-based SA using the constructed lexicon with different setting, only polarity words, polarity and negation words, polarity and emojis, and polarity along with both negation and emojis. The other experiment was performing machine learning-based SA with a variety of features, including different n-grams, emojis, lexicons, and negation. A summary of the paper's key contributions is as follows:

- Building of an ArSentiCOVID lexicon, a first lexical resource for Arabic SA about COVID-19.
- Developing a lexicon-based sentiment analyzer tool that can properly handle both negation and emoji.
- Constructing an extensive list of Arabic negation.
- Scraping a large Arabic corpus from Twitter about COVID-19.
- Annotating an Arabic sentiment corpus about COVID-19, a new Arabic reference corpus for SA, automatically annotated by based mainly on the constructed lexicon.
- Conducting an in-depth study using lexicon-based approach to investigate the usefulness and quality of the ArSentiCOVID lexicon.
- Introducing an ensemble method that combines lexicon-based sentiment features (negation, polarity, and emojis) as input features for a ML classifier to generate a more precise Arabic SA procedure.

The rest of the paper is organized as follows. In Section 2 we present a literature survey of Arabic lexicon construction and Arabic sentiment corpus annotation. In Section 3, we discuss our methodology for the lexicon construction, sentiment analyzer process, and corpus annotation. In Section 4, we present the experimental results and illustrate the conclusion and future work in Section 5.

2. Related Works

This section discusses the current research on Arabic SA with emphasis on sentiment lexicon and corpora annotation, and Arabic SA regarding COVID-19.

2.1. Arabic Lexicon Construction

Lexicons in SA include a group of words with their related polarity rather than a word and its definition like in dictionaries. The sentiment lexicon is considered the most crucial resource for most SA algorithms. There are three approaches to the construction of the sentiment lexicon, manual, automatic, and semi-automatic.

Manual lexicon construction requires human effort to extract sentiment words from Arabic dictionaries or from Arabic sentiment analysis datasets, and then label these words with their polarity. Mataoui et al. [13] concentrated on the Algerian dialect and the creation of a lexicon with a manual translation of the current MSA and Egyptian lexicon. The lexicon is divided into three sections, keywords, negation words, and intensification words. The authors subsequently expanded their work with two lexicons, one for emoticon vocabulary and the other for popular words. The Keyword Lexicon consisted of 2380 negative polarity words and 713 positive polarity words. The intensification words lexicon was built from MSA and then added the equivalent from the Algerian dialect. Experimental results demonstrate that their method performed well with an accuracy of 79.13%. The lexicon cannot handle emojis. Another work done by [14] developed a lexicon for sentiment analysis of Saudi dialect tweets called SauDiSenti, which consists of 4431 words and phrases that were manually identified by two Saudi annotators from previously annotated datasets of Saudi dialect tweets. They assessed SauDiSenti's performance using a new dataset labeled by two annotators, consisting of 1500 tweets uniformly dispersed across three classes, positive, neutral, and negative. The lexicon cannot handle negation or emojis.

Badra et al. [15] provided ArSEL, the first large scale Arabic sentiment and emotion dictionary. ArSEL is automatically generated by combining three lexical resources, DepecheMood, EWN, and ArSenL. At first, DepecheMood is mapped to EWN. Then it is iteratively enlarged utilizing the EWN synonymy semantic relation. Then the resulting extended DepecheMood version, EmoWordNet, is linked to ArSenL items using EWN synset IDs in both lexicons. ArSEL is made up of 32196 Arabic lemmas labelled concurrently with sentiment and emotion scores. The lexicon was evaluated using the SemEval 2007 Affective Task Arabic annotated dataset and the Arabic dataset from SemEval 2018 Task1. Coverages of 91% and 84% are obtained on both datasets, respectively. On the SemEval 2018 Arabic dataset, the average F1 metric for emotion classification improved by 30% when compared to the majority baseline. Another work carried out by Guellil et al. [16], where the lexicon is generated automatically by using the English dictionary "SOCAL". SOCAL has 6769 words with sentiment labels ranging from (1, 5) for negative words to (+1, +5) for positive words. Words were translated into Arabic, while their polarity remained unchanged. The dictionary has 4873 annotated terms ranging from (−5 to +5), with 2483 in Arabizi and 2390 in Arabic. The lexicon that was created was subsequently utilized to label an Algerian dataset. The suggested method generates automatically sentiment dataset of 8000 messages of which 4000 are allocated to Arabizi and 4000 to Arabic into positive and negative. The obtained F1-score is up to 72% for an Arabic dataset, while, for an Arabizi dataset, it is up to 78%. All previous lexicons could not handle negation.

Semi-automatic lexicon construction, the lexicon is built automatically and then manually reviewed. Al-Moslmi et al. [3] presented the Arabic senti-lexicon, a collection of 3880 positive and negative synsets labelled with their POS, dialect synsets, polarity scores, and inflected forms. The authors also developed a Multi-domain Arabic Sentiment Corpus (MASC) which contains 8860 positive and negative reviews from 15 domains. Two native speakers were instructed to identify these reviews with favorable and negative ratings written in various dialects in Arabic. Five popular machine learning classifiers are used, NB, KNN, SVM, LR, and neural networks are used as base classifiers for five different types of feature sets. The experimental results indicate that SVM produces the greatest results with Part Of Speech (POS) features. For the Saudi dialect, Assiri et al. [5] created an Arabic lexicon for the Saudi dialect SA. There are three main steps. In the first step, they utilized a learning algorithm that uses seed words and punctuation to extend the dictionary. In the second step, they employed a dictionary produced by Badaro et al. [17] and translated it

using the Buckwalter translation. This dictionary includes more than 150,000 words and associated punctuation. Finally, in the third step, they manually introduced new terms. It has more than 14,000 sentiment terms. The authors provided four rules for dealing with negation. The implementation of these rules improved performance by 3%.

To the best of our knowledge there are no research studies that have attempted to build a lexicon for Arabic SA that is devoted to the COVID-19 pandemic. Thus, the ArSentiCOVID lexicon is the first lexicon constructed for COVID-19. The developed lexicon handles both negation and emojis. The lexicon is large, containing 39,428 polarity words. Furthermore, it is taken advantages of both construction methods, manual and automatic.

2.2. Arabic COVID-19 Corpus

Since December 2019, social media platforms, and Twitter in particular, have proven to be indispensable for sharing information and discussing the COVID-19 pandemic. To facilitate the analysis of these discussions, a large volume of raw datasets were made available. The majority of the COVID-19 datasets are unlabeled tweet collections, such as as [12], that present a large Arabic tweet dataset that contain keywords relevant to COVID-19. The dataset was collected using the Tweepy Python library and Twitter streaming API from 1 January 2020 to 15 April 2020. Similarly, ArCOV-19 [10] is an Arabic dataset collected by searching Twitter over the course of one year from 27 January 2020. The dataset contains approximately 2.7 million tweets, along with the propagation networks of their most popular subset.

In addition to unlabeled data, ref. [18] manually annotated tweets in English (containing 504) and Arabic (containing 218). The dataset was annotated for various purposes, including harmfulness to society, the relevance of tweets to policymakers or governments, and fact-checking. The authors in [19] obtained a total of 1M unique Arabic tweets about COVID-19 from December 2019 to April 2020. The authors cluster tweets into five by using the K-means algorithm. The dataset was annotated for fact detection tasks. The authors in [11] annotate the 10-thousand English and Arabic dataset, called SenWave. The dataset annotated for fine-grained SA task. ArCovidVac is the largest manually annotated Arabic tweet dataset developed by [20] The dataset includes 10,000 tweets. The dataset annotated for various tasks consists of distinguishing informative tweets from uninformative ones; stance detection utilizing transformer architectures; and fine-grained multi-class tweet classification.

As a summary, the publicly available datasets are either unlabeled or manually annotated, whereas our dataset is automatically annotated using the COVID-19 lexicon. The dataset contains larger than 42K tweets annotated with the three labels positive, negative, and neutral.

3. Research Methodology

This section describes the two primary phases utilized to conduct this research. First, we present an illustration of the lexicon's sequential building phases. Second, we discuss the steps of the construction of sentiment analyzer tool to annotate our collected COVID-19 tweets.

3.1. Lexicon Construction

Previous approaches suffer from the high cost of developing a sentiment dictionary "from scratch", we have decided to take advantage of the lexicons that are already available and make improvements to them in order to establish our sentiment dictionary. To this end, we present a semi-automatic technique that makes use of two Arabic sentiment lexicons, the Arabic senti-lexicon [3] and the NileULex [21]. In addition, the approach includes automatic extraction of sentiment words from an annotated Arabic COVID-19 dataset called SenWave, as well as manual checking of all words in the proposed lexicon. Our strategy is organized into three stages, which include both automated and manual steps.

As seen in Figure 1, these stages include developing keyword and opinion rules lexicon, developing a lexicon using training data, and reforming and revising the lexicon.

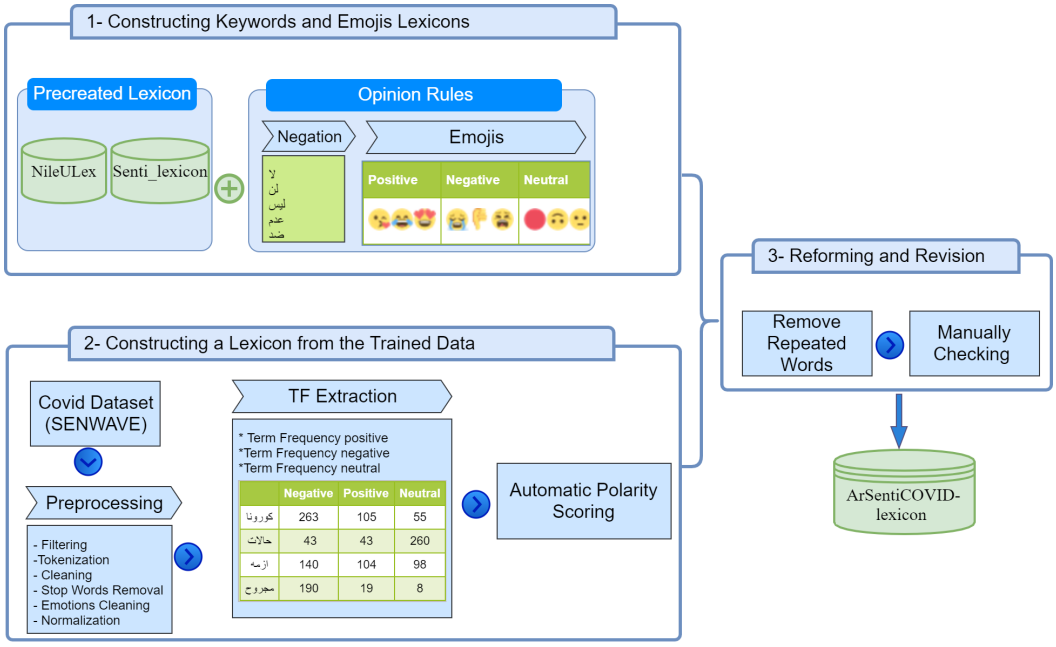


Figure 1. A proposed framework for lexicon construction.

3.1.1. Constructing Keyword and Opinion Rules Lexicons

The first phase in developing our Arabic lexicon for SA was to create two lexicons, a keyword lexicon and an opinion rules lexicon. To construct the keyword lexicon, we depend on two existing Arabic lexicons. The reasons behind the selection of these lexicons include their wide acceptance, the MSA coverage, as well as their relatively wide coverage. The first lexicon was NileULex, which had entries that were separated into compound phrases, single terms, or common idioms. We modified this lexicon to have only single words by deleting both common idioms and compound phrases. The second lexicon is the Arabic senti-lexicon, which has 2704 negative words and 1176 positive words written in MSA and annotated with their inflected forms, dialects synsets, POS, and polarity scores. Table 1 describes the statistics of the two used Arabic lexicons. We combine these two lexicons, then store all positive sentiment words in a separate file, and the same goes for all the words containing a negative sentiment. Finally, the keyword lexicon contained 2457 terms with a positive polarity and 6397 words with a negative polarity.

Table 1. Statistics of the pre-created lexicon.

	#Positive	#Negative	Total
NileULex	1281	3693	4974
Arabic senti-lexicon	1176	2704	3880

To improve the efficiency of our lexicon, we have constructed an opinion rules lexicon, an opinion rule is an implication with implied opinion on the right and an expression on the left. The expression is conceptual since it reflects a notion that can be represented in a variety of ways in a sentence. For example, "أنا لا أحب هذا الفيروس اللعين" "I do not like this damn virus" The word "not" changes the polarity of the word "like" from positive to negative. The developed opinion lexicon rules are composed of four lists: negation lists, positive emoji, negative emoji, and neutral emoji lists. Table show examples for each list.

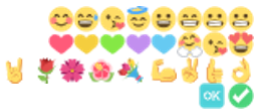
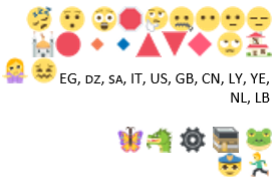
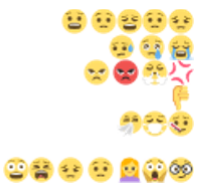
Negation words can change every word's sentiment polarity from negative to positive and from positive to negative. As such, we manually gathered the most widely accepted

negation terms used in MSA and Egyptian dialect, and then store them in a negation file which includes 34 Arabic words such as "ليس، دون، لا، لم".

On the other hand, Emojis are visual symbols, also known as ideograms, that represent not only face reactions but also ideas and concepts including vehicles, celebration, weather and buildings, foods and drink, animals and plants, or feelings, emotions, and activities. We measured the sentiments of emojis using negative, positive, and neutral values rather than a Likert scale. This is due to the difficulties of achieving intercoder reliability using a scalar technique, made much more challenging by the fact that the emojis we studied were uniquely associated with the COVID-19 pandemic.

We used the emojipedia (https://emojipedia.org, accessed on 4 January 2023) to classify emoji into its corresponding sentiments. The emojipedia is a centralized database of the majority of emojis available on modern smart devices, and it contains information about each emoji's numerous meanings, interpretations, Unicode, and usage. At first, as shown in Table 2, appropriate positive emojis were used to represent positive sentiment emotions. Smiles, laughter, love, and hugs all convey positive moods through facial expressions. Similarly, neutral sentiment emotion has been described in order to maintain appropriate neutral emojis. We have already looked at straight face, no expression, shock, flags, foods, buildings, animals, surprise, and indecision as examples of neutral facial expressions that can be analyzed. Negative sentiment emotion has been described to maintain relevant negative emojis. We have begun to study anger, shame, sadness, worry, disgust, and crying. Each positive emoji character was stored in a positive emoji list totalling 69 emojis. A neutral emoji character was saved in the neutral emoji list, totalling 199 emojis, while the negative emoji list included the emoji characters that have negative sentiment totalled 52.

Table 2. Sentiment and emotion expressed by emojis.

Emoji Group	Emoji	Emoji Sentiment	Number of Emojis (320)
Smileys Happiness Love		Positive	69
Straight face No expression Hesitation Surprise Shock Flags Animals Job		Neutral	199
Sadness Cry Anger Annoyed worry Disappointed Great dismay Horror Frowning		Negative	52

3.1.2. Constructing a Lexicon Using the Trained Data

In this phase, a labeled dataset called SenWave, which is the first available annotated dataset about COVID-19, is used to construct an Arabic sentiment lexicon automatically. This dataset was gathered from 1 March until 20 January 2020. It has been manually annotated and is available for download. Each tweet was annotated by at least three Arabic annotators with extensive expertise, all of whom were subjected to thorough quality control.

Table 3 contains instances of tweet text and its polarity culled from the SenWave corpus. The tweets in the corpus were primarily classified into four classes, negative, positive, joking, and neutral. In the dataset, there are 1418 joking tweets, however our suggested approach aims to identify only positive, neutral, and negative sentiments. As a result, in this study, we did not include any tweets having a joking sentiment. There are 1562 positive tweets in the dataset, 4269 neutral tweets, and 2750 negative tweets in the dataset. Table 4 provides statistical information about the SenWave dataset.

Table 3. Examples from SenWave dataset.

Tweet Text	Polarity
أنا متفائل ب كورونا إنه هيخلصنا من الظالمين إن شاء الله	Positive
صباح اول يوم في ابريل صباح اتنهاء فيروس كورونا باذن الله تعالى	Positive
باقي ١٠ ايام ويخلص الحجر وترجع الحياه وبموت كورونا	Positive
المواطن الصيني اللي اكل خفاش مما ادي لحمله لفيروس كرونا ومنه انتقل هذا الوباء اللعين للكرة الارضيه جمعاااء	Negative
يعنى لما اخد اجازه من الشغل و اانتخ تيجى كرونا تخلينى زى اللقيط معرفش اخرج من البيت	Negative
شفاء ٧ حالات ليصبح عدد الحالات التي تم شفاؤها من فايروس كورونا المستجد ٨٠ حالة	Neutral
اذا كان الكورونا مرض وابتلاء فإن نشر الشائعات وترويجها قرار خبيث او غباء محض !»»»	Neutral

Table 4. Statistics and measurements for the SenWave dataset used in this research.

Total tweets in original SenWave dataset	9999
Total Tweets after removing joking sentiment tweets	8581
Total number of positive tweets	1562
Total number of negative tweets	2750
Total number of neutral tweets	4269
Total number of words	122,005
Total number of characters	678,915
Average words per tweet	14.2

As earlier noted in Algorithm 1, the dataset was read from a file, and then many pre-processing steps were performed, as explained in Section 3.2.2. After preprocessing, all of the unique words extracted from the dataset's tweets were aggregated into a single set, known as a bag-of-words as explained in step two. In step three, the number of times each term appeared in tweets that were either positive, negative, or neutral was determined using the training dataset. After that, in step four, the polarity of each word was computed using the following formula, considering a labeled dataset with (L) as the length of each tweet, which comprises tweets of three categories, negative, positive, and neutral. According to line 19, to calculate the positive score of word w (PositiveScore(w)) divide the number of occurrences in positive tweets (#pos) by the total number of times it appears in all tweets. As stated in line 21, the neutral score of word w (NeutralScore(w)) is the frequency with which the term appears in tweets that are neutral (#neu) divided by the total occurrences of the term. As shown in line 20, the negative score of word w

(NegativeScore(w)) is the number of occurrences in negative tweets (#neg) divided by the total number of occurrences. Finally, in step five, in lines 22–29, we determine the final polarity of every word as follows, when the PositiveScore(w) is greater than both the NegativeScore(w) and the NeutralScore(w)), the term is considered positive. A word is considered to be negative if the NegativeScore(w)) is larger than both the PositiveScore(w)) and NeutralScore(w)) values; otherwise, its polarity is neutral.

Algorithm 1: Constructing a lexicon using the trained data.

Input : Preprocessed annotated corpus
Output: positive,negative, and neutral words

1 // Step 1: *Data Loading*;
2 $Preprocessed\_Annotated\_tweets \leftarrow Dataset.getAnnotatedTweets()$;
3 // Step 2: *create matrix of unique words*;
4 **foreach** *row* $\in$*Preprocessed_Annotated_tweets* **do**
5 $tweet\_words \leftarrow row(tweet\_text).split("")$;
6 $matrix.add(tweet\_words)$;
7 // Step 3: *Calculate the number of occurance of positive,negative,and neutral*;
8 **foreach** *word w in matrix* **do**
9 $\#pos = 0, \#neg = 0, \#neu = 0$;
10 **if** *w in Positive tweet* **then**
11 $\#pos = \#pos + 1$;
12 **end**
13 **if** *w in negative tweet* **then**
14 $\#neg = \#neg + 1$;
15 **else**
16 $\#neu = \#neu + 1$;
17 **end**
18 // Step 4: *calculate the score of positive ,negative,and neutral*;
19 $$PositiveScore = \frac{\sum_{1=1}^{L} \#pos(w,i)}{\sum_{1=1}^{L} \#pos(w,i)+\sum_{1=1}^{L} \#neg(w,i)+\sum_{1=1}^{L} \#neu(w,i)}$$
20 $$NegativeScore = \frac{\sum_{1=1}^{L} \#neg(w,i)}{\sum_{1=1}^{L} \#pos(w,i)+\sum_{1=1}^{L} \#neg(w,i)+\sum_{1=1}^{L} \#neu(w,i)}$$
21 $$NeutralScore = \frac{\sum_{1=1}^{L} \#neu(w,i)}{\sum_{1=1}^{L} \#pos(w,i)+\sum_{1=1}^{L} \#neg(w,i)+\sum_{1=1}^{L} \#neu(w,i)}$$
22 // Step 5: *Decide the sentiment polarity of every word* ;
23 **if** *positiveScore* $>$ *NegativeScore && positiveScore* $>$ *NeutralScore* **then**
24 *Word polarity* $\leftarrow$ *Positive*;
25 **end**
26 **if** *NegativeScore* $>$ *PositiveScore && NegativeScore* $>$ *NeutralScore* **then**
27 *Word polarity* $\leftarrow$ *Negative*;
28 **else**
29 *Word polarity* $\leftarrow$ *Neutral*;
30 **end**
31 **end**
32 **end**

We check the sentiment score of some words and presented the results in Table 5. For example, we found that the word "ازمه/crisis" appeared 140 times in the negative tweets, 98 times in neutral tweets, and 104 times in the positive tweets. In accordance with the formula, the negative score of the word "ازمه/crisis is: 140/(104 + 140 + 98) = 0.4, the neutral score is 0.2 while the positive score is 0.3. Based on the negative, positive, and neutral scores of the word, we can decide about its polarity. Thus, the word "ازمه/crisis carries a negative sentiment. we found that the word " العالم/world" appeared 5 times in the negative tweets, 67 times in neutral tweets, and 27 times in the positive tweets. In accordance with the formula, the neutral score of the word " العالم /world " is: 67/ (5 + 27 + 67) = 0.67, the negative score is 0.05 while the positive score is 0.27. We can decide the word's polarity by looking at its negative, positive, and neutral scores. So that, the word " العالم/world" holds a neutral sentiment.

We found that the word "الصبر patience" appeared 130 times in the positive tweets, 5 times in negative tweets, and 48 times in the neutral tweets. In accordance with the formula, the positive score of the word "الصبر patience" is: 130/ (130 + 5 + 48) = 0.67, the negative score is 0.05 while the neutral score is 0.26. Based on the negative, positive, and neutral score of the word we can decide about its polarity. Thus, the word "الصبر patience" carries a positive sentiment.

Table 5. Examples of sentiment scores of words.

Word	Positive_Score	Negative_Score	Neutral_Score
ازمه/crisis	104	140	98
العالم/world	27	5	67
الصبر/patience	130	5	48

3.1.3. Reforming and Revision

The tasks performed in this step are two-fold, duplicate checking as well as manual checking. When we execute the first task, we look for instances of duplicate words because we want to guarantee that all words contained in the lexicon are distinct. As depicted in Figure 2, the size of the positive file decreased by 861 words, the size of the neutral file decreased by 1063 words, and the size of the negative file decreased by 1744 words. As part of the second assignment, we undertake manual testing, which results in some words being located in a neutral file but having a negative sentiment, such as the word "الوباء لا الازمات الابتلاء الغباء الغبى الغلاء الفاجعه الفاسد". Some words contained in the negative file, on the other hand, have neutral polarity, such as the words "علاج شفاء الاقوياء الاقوي"". A number of words, such as "الناس" located in negative file however, they have neutral polarity. Although certain words are in the positive file, they having a negative sentiment, as in "ينفعك يهتم يحبون بالشفاء بالصحة بالعلاج
بالرحمه بندعمهم تحبه"". A small number of words stored in the positive file, but having neutral polarity "الجندي الجامعات الجوالات الحاسب".

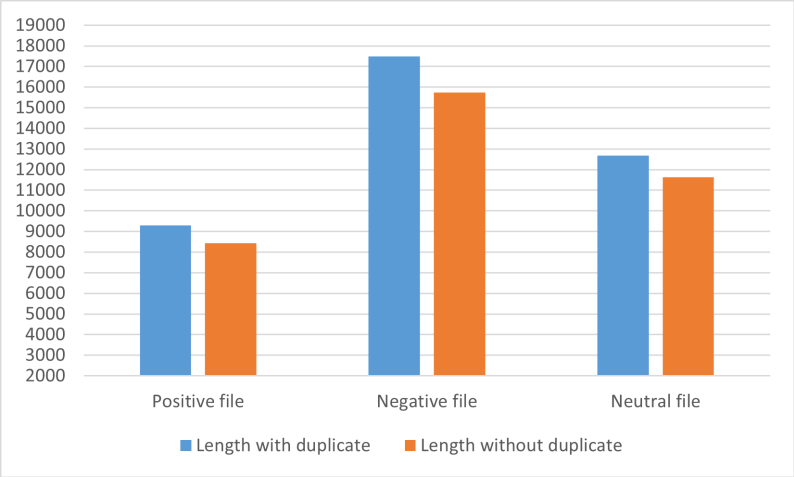


Figure 2. Size of positive, negative, and neutral file without/with duplicate.

3.2. Sentiment Analyzer

An efficient Arabic sentiment analyzer is developed for performing SA of Arabic tweets regarding to COVID-19 based on lexicon approach. This tool takes Arabic tweets as input, process them, and categorize them as neutral, positive, or negative based on lexicon approach and take into consideration negation, and emojis. To develop this tool several steps are performed, data preprocessing, sentiment score extraction, and sentiment computation. These steps will be described in detail as follows, as shown in Figure 3.

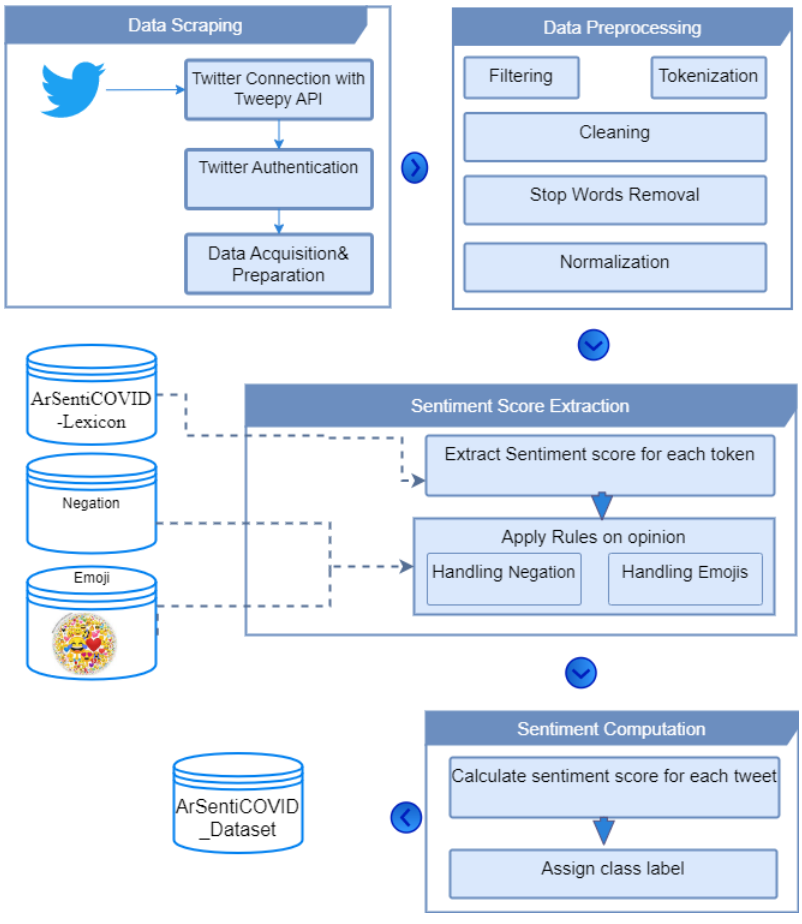


Figure 3. A proposed framework for sentiment analyzer.

3.2.1. Scraping Tweets

In general, the most important factor of a study is the data collected. This is because evaluation and experimentation are based on that data. Twitter has been chosen as the social media platform to collect datasets. Twitter makes its data available through public APIs that may access via URLs. For Twitter data collection, calling required libraries, such as Tweepy, is the first step to access Twitter data. Tweepy is a Python library that allows to access Twitter's data via the API. Then, an OAuth protocol-supported user authentication and a keep-alive HTTP connection method were utilized to gain access to the Twitter API. Specifically, the OAuth authentication protocol is utilized to grant applications permission to Twitter's services. Python function/API is utilized to fetch tweets from Twitter based on specified keywords, such as "كوفيد ١٩ ، فيروس كورونا، جائحة كورونا، وباء كورونا، فيروس كورونا المستجد، ازمة كورونا، اخر مستجدات فيروس كورونا، اخر تطورات كوفيد ١٩" and hashtags as "#كوفيد ١٩ ، # فيروس كورونا، #جائحة كورونا، #وباء كورونا، #فيروس كورونا المستجد، #اخر مستجدات فيروس كورونا، #اخر تطورات كوفيء" about COVID-19 We retrieved a total of 54,487 Arabic tweets under all keywords and hashtags from April to November in 2020. All the downloaded tweets are written in Arabic.

3.2.2. Data Preprocessing

In the Arabic SA, preprocessing is the most important stage, consisting of several cleaning and preparation techniques employed to the obtained data in order to render it more readable by humans and suitable for the classification stage. The suggested preprocessing steps and their impact on the input tweet are depicted in Figure 4. The details of steps performed in preprocessing are described below:

Filtering:in this step, two tasks are performed, which are removing repeated tweets appearing more than once and removing re-tweets. Re-tweeting is a common practice in Twitter in which the users republish or forward a tweet posted by another user and posting to another account with an indication that the source of the post is another user. Without the ability to filter out retweets, SA systems will only be presented with a flood of messages that have already been shared by other users. Therefore, removing re-tweets is a necessity for SA. Since the Twitter API frequently produces many tweets with the same content, we eliminated all duplicates by comparing each tweet to others.

Tokenization: means separating a string into individual elements, or tokens, like keywords, words, symbols, sentences, and other elements using white space and punctuation marks. In this work, the tweet was divided into individual words using spaces. In our experiment, tokenization was implemented using the NLTK library.

Tweet Cleaning: handles the noisy nature of Twitter data, cleaning it up by removing the extraneous stuff. For example, removing usernames, non-Arabic letters, images, and special characters like ($, %, &, _, -,#) from the tweet.

Normalization: two main tasks are performed. First, it removes repeated letters. For example, it replaces "كوروووونا" with "كورونا" by using a specific regular expression. The second task was substituting the same letters that are used interchangeably by one of them and make sure that no letters were taken out of the word; the conditions for normalization are as follows:

- Substitute "أ","إ" ,and "آ" for bare alif "ا" regardless of where in the word it appears.
- Substitute the final "ة" for "ه".
- Substitute the final "ى" for "ي".
- Substitute the final "ئ" and "ؤ" for "ء".

Stop Word Removal: Stop word is a group of words, such as prepositions, conjunctions, and articles, that do not affect the the text's meaning or provide any information. We remove stop words by using a list of stop words that is available on [22]. We omitted particular stop words that influence the ultimate sentiment of a tweet.

Figure 4. An example of preprocessing steps.

3.2.3. Sentiment Score Extraction

After the dataset was collected, several steps of preprocessing were performed. The words from each incoming tweet is scanned one by one in seven diverse files, negative word file, neutral word file, positive word file, positive emoji file, negative emoji file, neutral emoji file, and negation word file. Add one to the positive counter when a term is located in the positive word file. Add one to the negative counter if a word is located in the negative word file. Add one to the neutral counter when a term is located in the neutral word file. If a term is located in negation file or in emoji files, it is assigned polarity according to the following rules.

In the SA task, handling negation is a crucial stage because negation can influence the text's overall polarity [23]. In this study, we employ a more straightforward but efficient method for handling negation. In case a negation word is found, we will examine the polarity of the next token; if the next token has a positive polarity, then we add to the counter of negative words. However, if the word is negative, then this time we add to the counter of positive words. If the following token does not have any sentiment, then the next token will be examined and the counter will increase for negative or positive depending on the word's polarity. It is essential to specify the scope of negation, i.e., the word order in the sentence that may be modified by a negation term. In a preliminary experiment, we determined that a window size of three is effective for finding negated terms and according to [24] a window size of four yields excessive error rates. Consequently, we decided to examine the following three words at most.

We were able to assess the sentiment of each tweet by referring to the lists of positive, negative, and neutral emojis that had been provided. The technique is as follows. We started by converting text messages into Unicode representations, and then using regular expressions, we were able to extract Unicode characters that corresponded to the Unicode Consortium's emoji list. From our original dataset, we were able to identify a total of 316 distinct emoji characters. Afterwards, we compare each emoji using the following criteria. If the emoji is identified in the positive emoji file, it will be added to the positive counter by one. The negative counter is increased by one if an emoji from the negative emoji file is discovered. If an emoji is detected in a neutral word file, it will be added to the neutral counter as an additional one.

3.2.4. Sentiment Computation

To obtain the sentiment of an Arabic tweet we check the positive counter, negative counter, and neutral counter of the words of specific tweet as the following: Tweet is considered positive if its positive counter is larger than both its negative counter and the neutral counter. Tweet is considered negative if its negative counter is larger than both its positive counter and its neutral counter. otherwise, the tweet is considered as neutral

3.2.5. Dataset Description

A sample annotated dataset and their corresponding labels are shown in Table 6. To sum it up, there are 42,461 tweets in the dataset, with an average of 7.44 words per tweet and a total character count of 1,904,517 in the 315,936 words. For each sentiment class, Table 7 lists the statistics about the dataset.

Table 6. Sample of positive, neutral, and negative tweets.

Tweet	Label
RT @faisal_b_saud:اتباعك للتعليمات الصحية والتباعد الاجتماعي من أهم الأسباب بعد الله لحمايتك وحماية من تحب والمجتمع من فيروس #كورونا الجديد	Positive
التزامك بالبقاء في المنزل وعدم الخروج، واتباعك لتوجيهات وإرشادات الجهات الرسمية المختصة، واجب ديني وطاعة لولي #كورونا	Positive
RT @ahmedaljassim00:تتواصل الجهود الأمنية في تطبيق منع التجول استمرارا للجهود المبذولة في الحد من انتشار فيروس كورونا	Positive
RT @hassanafaa:تمكن فيروس كورونا الضعيف من محاصرة شعوب الأرض قاطبة في سابقة لا مثيل لها في تاريخ البشرية، ولأنه ليس بمقدور أحد التنبؤ بموع	Neutral
RT @hureyaksa:غضب كبير بين الأفارقة في #الصين بسبب عنصرية السكان المحليين واتهامهم بنشر فيروس #كورونا حيث تم طردهم من منازلهم ومن الأحياء	Neutral
هذا فيروس #كورونا له مدة وينقضي وهو خطير ووباء ولكن الحوادث تقتل اكثر منه ومن يظن انه سيقضي على الناس	Neutral
عاجل تسجيل حالتين وفاة بسبب فيروس #كورونا	Negative
كشف الوجه القبيح للغرب وأثبت أنه بالفعل مجرم قاتل لا يحترم حقوق الإنسان التي صدعنا بها بمنعه العلاج وأجهزة فيروس #كورونا	Negative
RT @Betkoen90:. عزل فيروس كورونا الناس و حجر عليهم في منازلهم	Negative

Table 7. Statistics on the tweets annotated dataset of Arabic COVID-19.

	Positive	Negative	Neutral
Total tweets	8672	5946	27,843
Total Words	63,885	46,195	205,856
Total Characters	363,710	276,724	1,264,083
Average words in each tweet	7.36	7.76	7.39
Average characters in each tweet	41.94	46.53	45.40

4. Experimentation and Simulation

4.1. Experimental Setup

4.1.1. Dataset Description

The authors in [25] construct and distribute an annotated Arabic dataset in context of COVID-19 called AraCOVID19-SSD. This dataset contains 5162 tweets that have been annotated for two purposes, SA and sarcasm identification. Tweets were collected between 15 December 2019 and 15 December 2020. All tweets in the dataset have been annotated and confirmed manually by human annotators. Only tweet-ids have been made available by the authors, thus we had to create a Python script to retrieve the tweet text of each tweet-id. Because 614 of the tweet-ids return no data, the size of the AraCOVID19-SSD dataset after hydration is equal to 4548 tweets. For each sentiment class, Table 8 lists the statistics about the dataset.

Table 8. Statistics about AraCOVID19-SSD dataset.

Total tweets in original AraCOVID19-SSD dataset	4548
Total number of positive tweets	1762
Total number of negative tweets	955
Total number of neutral tweets	1831
Total number of words	72,122
Total number of characters	493,356
Average words per tweet	15.85

4.1.2. Feature Representation

Features are a numerical representation of factors that are extracted from text and utilized as inputs to a specific machine learning classifier. Selecting appropriate features is crucial to the success of machine learning categorization. In this study, we examined the impact of combining TF-IDF with N-gram (unigram, bigram, trigram, fourgrm) technique to represent text. Furthermore, we define an extensive set of features as described in Table 9. We organized these features for three feature sets, sentiment-based features, emoji-based features, and negation-based features.

N-gram is one of the well-known models used in NLP fields and language modeling. N-gram is a model that consists of an adjacent sequence of elements (words, bytes, syllables, or characters) of N-length. In this work, Several baseline word-based n-gram features are tested. We also investigated how varying the length of n-grams impacts the accuracy of various classification classifiers.

TF-IDF is a numerical statistic model utilized to determine the significance of every word within the dataset. It is a well-known, straightforward method for extracting features that is employed in NLP, recommendation tasks, and classification. Two fundamental steps are required to implement this model. First, the Term Frequency (TF) It counts how many times a word appears in a tweet relative to the entire length of tweet words. It is stated as follows:

$$TF = \frac{NumberOfTimesTerm(t)AppearsInATweet}{NumberOfTermsInTheTweet} \quad (1)$$

Second, the Inverse Document Frequency (IDF) of uncommon words is high while that of common words is typically low. Hence, having the impact of highlighting unique words is explained as follows,

$$IDF = log(\frac{N}{n}) \quad (2)$$

where N is the number of tweets and n is the number of tweets a term t has appeared in. Then, to obtain the TF-IDF, we multiply the TF by each word's IDF value. The TF–IFD score increase proportionately with the frequency with which a word appears in the tweet

and decreases with the frequency with which the term appears in the corpus. Commonly occurring terms in the corpus have lower TF-IDF values [26].

Sentiment-based feature: sentiment words are words in a language that are used to express positive or negative sentiments. Examples of positive opinion words are 'شفاء (healing), 'فرح (joy), and ' تعافى ' (recover). Examples of negative opinion words are 'بلاء (scourge), 'جائحة ' (pandemic), and 'ابتلاء ' (plagued). The frequency of sentiment words is the simplest approach to representing a tweet with a vector. The frequency of sentiment words is the frequency with which each sentiment word appears in a tweet. We defined three features in this feature set, the frequency of positive words, the frequency of neutral words, and the frequency of negative words.

Emoji based Feature: using the frequency with which certain emojis appear in the dataset, we were able to derive three features, the frequency of positive emojis, the frequency of neutral emojis, and the frequency of negative emojis found in each tweet.

Negation: we employed two features, namely Is_Negation a binary feature indicates that there is a negation in the sentence and NoNegation feature to express how many negation terms appear in the sentences.

Table 9. Features extracted for each tweet.

Feature Set Name	Feature Name
Baseline model	Unigram
Sentiment based feature	F1. The frequency of positive words
	F2. The frequency of negative words
	F3. The frequency of neutral words
	F4. F1 + F2 + F3
Emoji based features	F5. The frequency of positive emojis
	F6. The frequency of negative emojis
	F7. The frequency of neutral emojis
	F8. F5 + F6 + F7
Negation based features	F9. The frequency of negation words
	F10. Absence/presence of negation
	F11. F9 + F10
All Features	F12. F4 + F8 + F11

4.1.3. Classification Algorithms

The next step, after the text turned into a set of features, is to pass the features into a machine learning algorithm to produce a model capable of classifying new data. In this work we used eight ML classifiers, SVM, Gaussian Naïve Bayes (GNB), Logistic Regression (LR), Multinomial Naïve Bayes (MNB), K Nearest Neighbor (KNN), Bernoulli Naïve Bayes (BNB), DT, and RF.

SVM is a linear classifier proposed by [27] that builds a model for predicting the class of the dataset. It deals with the problem by locating a hyperplane (boundary) with the greatest margin.

KNN is a case-based learning classifier [28] that works on the assumption that similar data are nearby. It computes the similarities between datasets using a typical similarity function (like Cosine similarity). After that, the new data are assigned to the same class as its nearest K neighbor.

NB predicts the class of new input data by computing the probability using Bayes' theorem. The class with the highest probability score for the provided input data is assigned to the new input. There are three models of NB: Gaussian, Bernoulli, and multinomial models. When the dataset is large, multinomial can be the right choice for the classification task. Equation (3) represents the Bayes theorem, where A denotes the class and B the data [26].

$$P(B) = \frac{(P(A) * P(A))}{(P(B))} \tag{3}$$

DT [29] is a likelihood classifier based on multistage of logical decisions. Moreover, it classifies an unlabeled document to its class based on several decision functions.

RF is a collection of tree structure classification models such that h (N, θ n), M = 1,2,3,...T, where N is the input, T is number of tree structure classifiers and θ is the independent random vectors. The general definition of RF classifier has been defined by [30]. Many other researchers provide different approaches such as [31,32] in which the difference is the way of obtaining the distributed random vectors. Every tree is working as a decision tree that selects the data randomly from the available data.

LR is a discriminative classifier. LR has been introduced by [33] for modeling categorical outcome of binary or multi-values variables. It solves the problem by extracting weighted features, in order to produce the label which maximizes the probability of event occurrence.

4.1.4. Evaluation Metrics

In the scope of this research, the performance of different used machine learning algorithms has been evaluated using the following frequently metrics, accuracy, recall, precision, and F-score. Accuracy is defined as the ratio of accurately predicted observations to total observations in Equation (4), where TP, FP, TN, and FN denote True Positive, False Positive, True Negative, and False Negative.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (4)$$

Precision is calculated as stated in Equation (5), which equals the number of correctly predicted positive occurrences divided by the total number of predicted positive occurrences.

$$Precision = TP/(TP + FP) \quad (5)$$

Recall is calculated as indicated in Equation (6), which is the ratio of correctly anticipated positive occurrences to the total number of positive opinions.

$$Recall = TP/(TP + FN) \quad (6)$$

Continuing with recall and precision, the F-Score can be calculated as harmonic mean of precision and recall, as illustrated in Equation (7).

$$F - score = 2 * Precision * Recall/(Precision + Recall) \quad (7)$$

4.2. Experimental Results

This section summarizes the findings from all the experiments conducted. Two experiments were conducted to determine the lexicon's efficiency, lexicon-construction approach results, ML classification models results. In the first experiment we used a lexicon-based approach to perform a three-way classification on the dataset. In the second experiment, different groups of features, sizes of N-grams varying from one to four, negation, emojis, and sentiment, were used to extract features, which were then evaluated using eight ML classifiers, SVM, KNN, NB, MNB, BNB DT, RF, and LR. It was decided that 80% of the dataset would be used for training, and the remaining 20% would be used for testing.

4.2.1. Lexicon Construction Approach Results

To assess the efficiency of the constructed lexicon, we carried out a series of experiments using a straightforward lexicon-based approach that does not require training or tuning. Three-way classification was applied to the datasets (positive, neutral, or negative). The purpose of this research is to determine whether negation detection and emoji handling have an impact on SA and classification accuracy and how they can be improved. As a result, we propose that the generated lexicons be evaluated in four different settings, 1—with and without negation handling, 2—with and without emoji handling, 3—with negation and emojis handling, and 4—without any rules (negation, emojis), respectively. Additionally,

we compare the performance of the pre-created lexicons (NileULex, senti-lexicon) to that of a lexicon generated using the same settings.

Table 10 displays the results of the four measures, accuracy, recall, precision, and F-score, obtained after applying the SA to the test corpus set using a lexicon-based approach. The results of two lexicons, NileULex and senti-lexicon, are provided in comparison of the constructed Arabic sentiment lexicon (ArSentiCOVID). It is observed that the performance of NileULex is comparable to that of senti-lexicon when no rules are applied. The ArSentiCOVID lexicon performs better than the other two lexicons. The NileULex lexicon improves significantly when negation rules are applied. The accuracy of senti-lexicon also improved when applying negation rules (55.46%). When applying negation rules, the ArSentiCOVID performs best, and improved by approximately 4.13%. When emoji rules were applied, ArSentiCOVID improved by approximately 2.13%. The developed lexicon improved by 6.13% after applying the negation and emoji rules. In general, it outperformed the other two lexicons.

Table 10. Results obtained with or without using negation rules.

Lexicon	Setting	Accuracy	Precision	Recall	F-Score
NileULex	Average without Applying Any Rule	40.27%	40.27%	40.27%	40.27%
senti-lexicon		40.27%	40.27%	40.27%	40.27%
ArSentiCOVID		76.87%	76.87%	76.87%	76.87%
NileULex	Average with Applying Negation Rules	55.57%	55.57%	55.57%	55.57%
senti-lexicon		55.46%	55.46%	55.46%	55.46%
ArSentiCOVID		81.00%	81.00%	81.00%	81.00%
NileULex	Average with Applying Emoji Rule	40.27%	40.27%	40.27%	40.27%
senti-lexicon		40.27%	40.27%	40.27%	40.27%
ArSentiCOVID		79.00%	79.00%	79.00%	79.00%
NileULex	Average with Applying Negation and Emoji Rules	55.57%	55.57%	55.57%	55.57%
senti-lexicon		55.46%	55.46%	55.46%	55.46%
ArSentiCOVID		83.00%	83.00%	83.00%	83.00%

4.2.2. ML Classification Models Results

These experiments were designed to assess and synthesize the quality and utility of developed language resources for Arabic SA, specifically the Arabic sentiment lexicon, as well as to develop a more precise SA tool. To accomplish this, it was necessary to first determine the classifiers' overall performance on Arabic SA without the use of an Arabic sentiment lexicon. In the beginning, the KNN, SVM, GNB, MNB, BNB, LR, RF, and DT classifiers were trained and tested using a simple model for classification (unigram feature). Table 11 summarizes the experimental results obtained with the eight classifiers without using any features. the RF and SVM algorithms achieve the highest accuracy of 89.01%, the highest precision of 89.51%, the highest recall of 89.01%, the highest F-score of 88.89%, but worst performance with the GNB classifier.

Table 11. Performance of ML classifiers for baseline model.

Feature	Classifier	Accuracy	Precision	Recall	F-Score
Unigram	SVM	89.01	89.51	89.01	88.86
	GNB	59.89	74.43	59.89	62.03
	MNB	87.91	88.13	87.91	87.74
	BNB	79.67	80.60	79.67	79.11
	DT	81.87	81.78	81.87	81.81
	RF	89.01	89.13	89.01	88.89
	LR	87.36	88.11	87.36	87.11
	KNN	73.08	79.79	73.08	69.92

Following that, we performed a comparative evaluation of lexicon-based features by contrasting the efficacy of different feature sets. Each of the eight machine learning classifiers (KNN, SVM, GNB, MNB, BNB, LR, RF, and DT) was conducted individually to study the impact and significance of each feature on sentiment classification. Mainly, we were trying to figure out which features in the Arabic ArSentiCOVID lexicon are most important for improving Arabic SA. This was achieved by employing eight machine learning classifiers to conduct several experiments on each feature and feature set individually.

Table 12 displays the overall performance of the eight classifiers on the Arabic SA task when each feature (bigram, trigram, and four-gram) is used. The SVM classifier achieved the best performance (89.01%) when the bigram feature is used, and the same result is obtained when the trigram feature is used, but the accuracy is reduced when the fourgram feature is used (86.26%).

Table 12. Performance of ML classifiers for N-gram feature.

Feature	Classifier	Accuracy	Precision	Recall	F-Score
bigram	SVM	89.01	89.76	89.01	88.82
	GNB	66.48	78.03	66.48	68.67
	MNB	88.46	88.86	88.46	88.33
	BNB	79.12	81.64	79.12	78.28
	DT	85.16	85.18	85.16	84.94
	RF	86.81	86.94	86.81	86.70
	LR	85.71	87.10	85.71	85.30
	KNN	67.03	76.93	67.03	61.74
trigram	SVM	89.01	89.76	89.01	88.82
	GNB	68.68	80.59	68.68	70.80
	MNB	87.91	88.40	87.91	87.76
	BNB	76.37	79.69	76.37	75.17
	DT	83.52	83.59	83.52	83.3
	RF	87.36	87.73	87.36	87.22
	LR	84.62	86.31	84.62	84.14
	KNN	65.38	76.31	65.38	59.50
fourgram	SVM	86.26	87.27	86.26	85.98
	GNB	68.68	80.59	68.68	70.80
	MNB	87.36	87.92	87.36	87.18
	BNB	76.37	80.58	76.37	74.77
	DT	84.07	84.03	84.07	83.92
	RF	86.26	86.53	86.26	86.09
	LR	82.97	85.17	82.97	82.37
	KNN	64.84	78.99	64.84	58.44

Table 13 compares the performance of the eight classifiers on the Arabic SA task when each feature (F1–F4) from sentiment-based feature set is used independently and when all of these features are used collectively. When using the feature F1 (the frequency of positive words), the SVM achieve the highest performance. When applying the F2 (the frequency of negative words) is used, the two classifiers SVM and LR achieve the highest performance. When using the feature F3 (the frequency of neutral words), the SVM achieve the highest performance. When all features are combined as in F4, the accuracy is increased by 2%. Based on these results, it was determined that the sentiment-based features are suitable for sentiment classification when used individually or in combination.

Table 13. Performance of ML classifiers for a polarity-based feature set.

Feature	Classifier	Accuracy	Precision	Recall	F-Score
F1	SVM	88.46	88.55	88.46	88.37
	GNB	59.34	73.46	59.34	61.45
	MNB	78.02	81.88	78.02	76.51
	BNB	80.22	80.85	80.22	79.70
	DT	87.91	87.90	87.91	87.89
	RF	86.81	86.67	86.81	86.64
	LR	85.71	85.96	85.71	85.41
	KNN	83.52	83.28	83.52	83.19
F2	SVM	90.11	90.45	90.11	89.92
	GNB	59.34	73.46	59.34	61.45
	MNB	86.26	86.95	86.26	86.05
	BNB	80.77	81.61	80.77	80.22
	DT	85.16	85.09	85.16	85.01
	RF	89.01	89.20	89.01	88.84
	LR	90.11	90.45	90.11	89.92
	KNN	84.07	84.56	84.07	83.68
F3	SVM	89.56	89.97	89.56	89.43
	GNB	59.34	73.46	59.34	61.45
	MNB	82.97	85.33	82.97	81.86
	BNB	79.12	79.87	79.12	78.52
	DT	84.62	84.65	84.62	84.43
	RF	88.46	88.48	88.46	88.33
	LR	87.91	88.55	87.91	87.68
	KNN	82.97	83.18	82.97	82.66
F4	SVM	92.86	92.93	92.86	92.86
	GNB	80.52	81.25	80.52	80.28
	MNB	88.31	88.97	88.31	88.45
	BNB	85.71	85.69	85.71	85.64
	DT	86.36	86.5	86.36	86.41
	RF	89.61	89.67	89.61	89.62
	LR	90.91	90.92	90.91	90.89
	KNN	88.31	88.51	88.31	88.35

Table 14 illustrates the overall performance of the eight classifiers on the Arabic SA task when each feature from the emoji-based (second feature set) is used separately or combined. When using the feature F5 (the frequency of positive emojis), the SVM classifier achieves the best results. When the feature F6 (the frequency of negative emojis) is employed, the SVM and RF classifier have the highest accuracy (89.01%). The SVM classifier achieves the best performance when the feature F7 (the frequency of neutral emojis) is utilized. When all features are merged, SVM achieves the best performance. These findings led to the conclusion that the features in this feature set (based on emojis) are suitable for sentiment categorization when employed singly or in combination. This means that simply counting the number of emojis in a tweet might give you a reasonable idea of its overall sentiment.

Table 15 displays the overall performance of the eight classifiers on the Arabic SA task when each feature from negation-based (the third feature set) is used separately or combined. When the feature F9 (the frequency of negation words) is utilized, the SVM classifier achieves the best performance. When the feature F10 (the presence/absence of negation word) is employed, the SVM classifier achieves the best performance. It is worthwhile to note that when all features are taken into account SVM provides the best results. From previous findings, it was inferred that the features in the negation-based feature set, whether used alone or combined, are suitable for the sentiment classification. This suggests that only presence/absence of negation word in a tweet is a reliable predictor of its overall sentiment.

Table 14. Performance of ML classifiers for emojis-based feature set.

Feature	Classifier	Accuracy	Precision	Recall	F-Score
F5	SVM	90.66	90.79	90.66	90.55
	GNB	59.34	73.46	59.34	61.45
	MNB	89.56	89.60	89.56	89.37
	BNB	81.87	82.43	81.87	81.54
	DT	85.16	85.13	85.16	85.07
	RF	89.56	89.67	89.56	89.55
	LR	86.81	87.13	86.81	86.50
	KNN	78.57	81.39	78.57	75.92
F6	SVM	89.01	89.33	89.01	88.84
	GNB	59.34	73.46	59.34	61.45
	MNB	87.91	88.12	87.91	87.79
	BNB	80.22	81.11	80.22	79.67
	DT	83.52	83.34	83.52	83.35
	RF	89.01	89.21	89.01	88.95
	LR	87.36	87.92	87.36	87.09
	KNN	76.92	82.22	76.92	75.35
F7	SVM	89.01	89.33	89.01	88.84
	GNB	59.34	73.46	59.34	61.45
	MNB	88.46	88.7	88.46	88.19
	BNB	79.67	80.60	79.67	79.11
	DT	82.42	82.32	82.42	82.34
	RF	86.26	86.27	86.26	86.05
	LR	87.36	87.92	87.36	87.09
	KNN	75.27	78.47	75.27	72.8
F8	SVM	89.61	90.08	89.61	89.69
	GNB	79.87	80.72	79.87	79.66
	MNB	87.01	87.34	87.01	87.02
	BNB	84.42	84.39	84.42	84.39
	DT	88.31	88.36	88.31	88.31
	RF	88.96	88.94	88.96	88.94
	LR	84.42	85.63	84.42	84.58
	KNN	68.18	71.93	68.18	67.28

Table 15. Performance of ML classifiers for negation-based feature set.

Feature	Classifier	Accuracy	Precision	Recall	F-Score
F9	SVM	89.01	89.51	89.01	88.86
	GNB	59.89	74.43	59.89	62.03
	MNB	87.91	88.13	87.91	87.74
	BNB	79.67	80.6	79.67	79.11
	DT	81.87	81.74	81.87	81.71
	RF	87.91	88.04	87.91	87.69
	LR	87.36	88.11	87.36	87.11
	KNN	73.08	79.79	73.08	69.92
F10	SVM	90.26	90.8	90.26	90.31
	GNB	79.87	80.84	79.87	79.68
	MNB	85.71	86.16	85.71	85.80
	BNB	85.71	85.66	85.71	85.65
	DT	86.36	86.35	86.36	86.35
	RF	88.96	88.99	88.96	88.97
	LR	87.66	88.35	87.66	87.75
	KNN	68.83	74.54	68.83	67.16

Table 15. *Cont.*

Feature	Classifier	Accuracy	Precision	Recall	F-Score
F11	SVM	90.26	90.80	90.26	90.31
	GNB	79.87	80.84	79.87	79.68
	MNB	85.71	86.16	85.71	85.80
	BNB	85.71	85.66	85.71	85.65
	DT	84.42	84.38	84.42	84.39
	RF	88.31	88.30	88.31	88.29
	LR	87.66	88.35	87.66	87.75
	KNN	68.83	74.54	68.83	67.16

Table 16 shows the results of the experiments (four performance parameters, Accuracy, Precision, Recall, and F-score) for all eight machine learning classification algorithms in predicting sentiments of tweets using all previously feature sets. Each of the SVM, RF, and LR models have the first rank with a greater accuracy of (92.21%). While the DT come in the second rank with accuracy of (90.26%). The third rank was associated with the KNN with accuracy of 88.31%. The fourth rank was Bernoulli NB with accuracy of 86.36%. The fifth rank come with Multinomial NB with accuracy of 85.71%. The last rank come with Gaussian NB with accuracy of 80.52%.

Table 16. Performance of ML classifiers for all feature sets.

Feature	Classifier	Accuracy	Precision	Recall	F-Score
F12	SVM	92.21	92.32	92.21	92.23
	GNB	80.52	81.25	80.52	80.28
	MNB	85.71	86.91	85.71	85.78
	BNB	86.36	86.34	86.36	86.27
	DT	90.26	90.34	90.26	90.25
	RF	92.21	92.32	92.21	92.23
	LR	92.21	92.24	92.21	92.22
	KNN	88.31	88.40	88.31	88.34

As seen in Figure 5, the use of all features considerably increased the accuracy of sentiment classification throughout all assessed classifiers. SVM achieves the highest accuracy (92.86%) when employing sentiment-based features, followed by LR (90.91%). When applying emoji-based features, the SVM achieves the highest accuracy (89.61%), followed by RF (88.96%) and LR (88.96%). The use of emoji-based characteristics dramatically enhanced the accuracy of sentiment classification on all classifiers investigated. When applying negation-based features, SVM achieves the maximum accuracy (90.26%), followed by RF (88.31%). As the best-performing classification classifier, SVM improved sentiment classification accuracy by roughly 3.85% when using the sentiment-based features, 0.6% when using the emoji-based features, 1.25% when using the negation-based features, and 3.2% when combining all features.

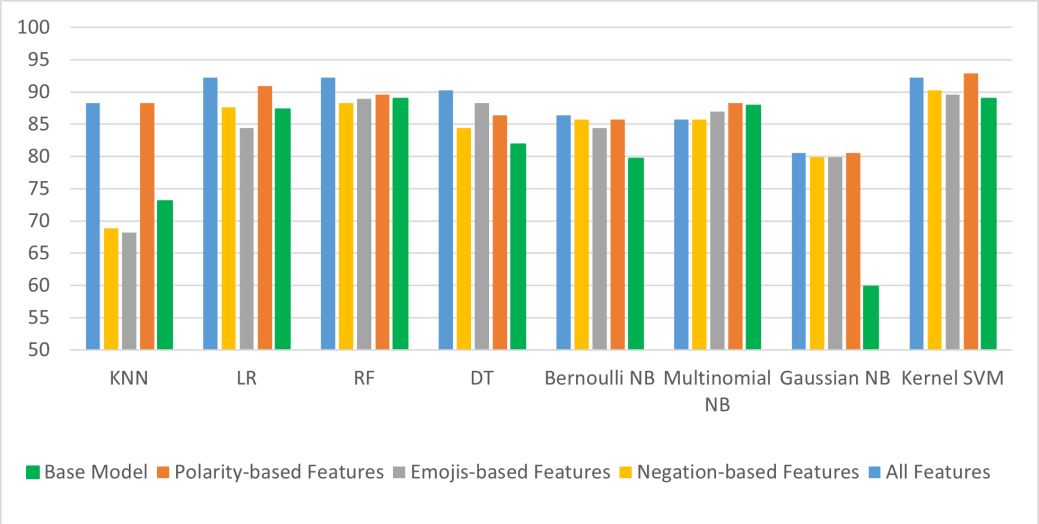


Figure 5. Accuracy measure for ML classifiers using different features.

5. Conclusions and Future Work

In this paper, we built ArSentiCOVID a large-scale Arabic sentiment lexicon about COVID-19 that took advanced linguistic phenomena like negation and emojis into account. The lexicon is developed both manually and automatically. The automatic step is articulated using two main methods, using the pre-created two lexicon and extract words from annotated dataset about COVID-19. For the manual step, we extracted and hand-labeled three extensive word lists, negation and emoji and word list from SenWave dataset. In addition, we developed a sentiment analyzer tool based on a lexicon that is capable of appropriately handling negation and emoji. As a result, we created a large-scale Arabic annotated sentiment corpus about COVID-19 collected from Twitter and contain more than 42000 tweets. To test the efficiency of the developed lexicon we run two experiments, using the AraCOVID19-SSD dataset. For the first experiment we test the effect of applying negation and emoji rules or both in the created lexicon by employing a lexicon-based approach to conduct a three-way classification on the dataset. The accuracy of the lexicon is improved by 2.13% when emoji rules are used, 4.13% when negation rules are applied, and 6.13% when both are applied. For the second experiment, we applied an ensemble method that uses lexicon-based sentiment polarity, negation, and emojis as input features for the ML approach. The results of the second experiment reveal that using sentiment-based features, emoji-based features, negation-based features, and all features with the SVM classifier improved opinion classification accuracy by about 3.85%, 0.6%, 1.25%, and 3.2%, respectively. Future work will include assigning sentiment scores to the lexicon entries. Extending the lexicon and experimenting with different types of datasets. Scraping more data about COVID-19 from different social networking websites. Both the lexicon and corpus are available upon request.

Author Contributions: Methodology, N.A. and T.H.A.S.; software, N.A.; writing—original draft, N.A.; writing—review and editing, I.E.E., M.F.F. and T.H.A.S.; supervision, I.E.E., M.F.F. and T.H.A.S.; formal analysis, N.A., I.E.E., M.F.F. and T.H.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data will be made available on reasonable request.

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

References

1. Guellil, I.; Azouaou, F.; Mendoza, M. Arabic sentiment analysis: Studies, resources, and tools. *Soc. Netw. Anal. Min.* **2019**, *9*, 56. [CrossRef]
2. El-Beltagy, S.R.; Ali, A. Open issues in the sentiment analysis of Arabic social media: A case study. In Proceedings of the 2013 9th International Conference on Innovations in Information Technology (IIT), IEEE, Al Ain, United Arab Emirates, 17–19 March 2013; pp. 215–220.
3. Al-Moslmi, T.; Albared, M.; Al-Shabi, A.; Omar, N.; Abdullah, S. Arabic senti-lexicon: Constructing publicly available language resources for Arabic sentiment analysis. *J. Inf. Sci.* **2018**, *44*, 345–362. [CrossRef]
4. Ballesteros, M.; Francisco, V.; Díaz, A.; Herrera, J.; Gervás, P. Inferring the scope of negation in biomedical documents. In Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, New Delhi, India, 11–17 March 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 363–375.
5. Assiri, A.; Emam, A.; Al-Dossari, H. Towards enhancement of a lexicon-based approach for Saudi dialect sentiment analysis. *J. Inf. Sci.* **2018**, *44*, 184–202. [CrossRef]
6. Alharbi, O. Negation Handling in Machine Learning-Based Sentiment Classification for Colloquial Arabic. *Int. J. Oper. Res. Inf. Syst. (IJORIS)* **2020**, *11*, 33–45. [CrossRef]
7. Al-Twairesh, N.; Al-Khalifa, H.; Al-Salman, A. Arasenti: Large-scale twitter-specific Arabic sentiment lexicons. In Proceedings of the Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 697–705.
8. Gamal, D.; Alfonse, M.; El-Horbaty, E.S.M.; Salem, A.B.M. Twitter benchmark dataset for Arabic sentiment analysis. *Int. J. Mod. Educ. Comput. Sci.* **2019**, *11*, 33. [CrossRef]
9. Al-Laith, A.; Shahbaz, M.; Alaskar, H.F.; Rehmat, A. AraSenCorpus: A Semi-Supervised Approach for Sentiment Annotation of a Large Arabic Text Corpus. *Appl. Sci.* **2021**, *11*, 2434. [CrossRef]
10. Haouari, F.; Hasanain, M.; Suwaileh, R.; Elsayed, T. Arcov-19: The first arabic COVID-19 twitter dataset with propagation networks. *arXiv* **2020**, arXiv:2004.05861.
11. Yang, Q.; Alamro, H.; Albaradei, S.; Salhi, A.; Lv, X.; Ma, C.; Alshehri, M.; Jaber, I.; Tifratene, F.; Wang, W.; et al. SenWave: Monitoring the global sentiments under the COVID-19 pandemic. *arXiv* **2020**, arXiv:2006.10842.
12. Alqurashi, S.; Alhindi, A.; Alanazi, E. Large arabic twitter dataset on COVID-19. *arXiv* **2020**, arXiv:2004.04315.
13. Mataoui, M.; Zelmati, O.; Boumechache, M. A proposed lexicon-based sentiment analysis approach for the vernacular Algerian Arabic. *Res. Comput. Sci.* **2016**, *110*, 55–70. [CrossRef]
14. Al-Thubaity, A.; Alqahtani, Q.; Aljandal, A. Sentiment lexicon for sentiment analysis of Saudi dialect tweets. *Procedia Comput. Sci.* **2018**, *142*, 301–307. [CrossRef]
15. Badaro, G.; Jundi, H.; Hajj, H.; El-Hajj, W.; Habash, N. Arsel: A large scale arabic sentiment and emotion lexicon. *OSACT* **2018**, *3*, 26.
16. Guellil, I.; Adeel, A.; Azouaou, F.; Hussain, A. Sentialg: Automated corpus annotation for algerian sentiment analysis. In Proceedings of the International Conference on Brain Inspired Cognitive Systems, Xi'an, China, 7–8 July 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 557–567.
17. Badaro, G.; Baly, R.; Hajj, H.; Habash, N.; El-Hajj, W. A large scale Arabic sentiment lexicon for Arabic opinion mining. In Proceedings of the EMNLP 2014 workshop on arabic natural language processing (ANLP), Doha, Qatar, 25–29 October 2014; pp. 165–173.
18. Alam, F.; Shaar, S.; Dalvi, F.; Sajjad, H.; Nikolov, A.; Mubarak, H.; Martino, G.D.S.; Abdelali, A.; Durrani, N.; Darwish, K.; et al. Fighting the COVID-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society. *arXiv* **2020**, arXiv:2005.00033.
19. Alsudias, L.; Rayson, P. COVID-19 and Arabic Twitter: How can Arab world governments and public health organizations learn from social media? In Proceedings of the Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Online, 9 July 2020.
20. Mubarak, H.; Hassan, S. Arcorona: Analyzing arabic tweets in the early days of coronavirus (COVID-19) pandemic. *arXiv* **2020**, arXiv:2012.01462.
21. El-Beltagy, S.R. Nileulex: A phrase and word level sentiment lexicon for egyptian and modern standard arabic. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portoroz, Slovenia, 23–28 May 2016; pp. 2900–2905.
22. Abdulla, N.A.; Ahmed, N.A.; Shehab, M.A.; Al-Ayyoub, M. Arabic sentiment analysis: Lexicon-based and corpus-based. In Proceedings of the 2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT), Amman, Jordan, 3–5 December 2013; pp. 1–6.
23. Kolchyna, O.; Souza, T.T.; Treleaven, P.; Aste, T. Twitter sentiment analysis: Lexicon method, machine learning method and their combination. *arXiv* **2015**, arXiv:1507.00955.
24. Ihnaini, B.; Mahmuddin, M. Valence Shifter Rules for Arabic Sentiment Analysis. *Int. J. Multidiscip. Sci. Adv. Technol.* **2020**, *1*, 167–184.
25. Ameur, M.S.H.; Aliane, H. AraCOVID19-SSD: Arabic COVID-19 Sentiment and Sarcasm Detection Dataset. *arXiv* **2021**, arXiv:2110.01948.

26. Aljabri, M.; Chrouf, S.M.B.; Alzahrani, N.A.; Alghamdi, L.; Alfehaid, R.; Alqarawi, R.; Alhuthayfi, J.; Alduhailan, N. Sentiment analysis of Arabic tweets regarding distance learning in Saudi Arabia during the COVID-19 pandemic. *Sensors* **2021**, *21*, 5431. [CrossRef] [PubMed]
27. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the European Conference on Machine Learning, Chemnitz, Germany, 21–23 April 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 137–142.
28. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In *On the Move to Meaningful Internet Systems, Proceedings of the OTM Confederated International Conferences*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 986–996.
29. Swain, P.H.; Hauska, H. The decision tree classifier: Design and potential. *IEEE Trans. Geosci. Electron.* **1977**, *15*, 142–147. [CrossRef]
30. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
31. Kwok, S.W.; Carter, C. Multiple decision trees. In *Machine Intelligence and Pattern Recognition*; Elsevier: Amsterdam, The Netherlands, 1990; Volume 9, pp. 327–335.
32. Dietterich, T.G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach. Learn.* **2000**, *40*, 139–157. [CrossRef]
33. Hosmer, D.W.; Jovanovic, B.; Lemeshow, S. Best subsets logistic regression. *Biometrics* **1989**, *45*, 1265–1270. [CrossRef]