Investigating the Impact of Utilizing the K-Nearest Neighbor and Levenshtein Distance Algorithms for Arabic Sentiment Analysis on Mobile Applications

Ahmed A. Al-Shalabi[1,*], Ghaleb Al-Gaphari [1], Salah AL-Hagree [1,2,*] , and Fahd Alqasemi[3]
[1] Computer Science Department, Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen.
[2] Computer Science Department, Faculty of Sciences, Ibb University, Ibb, Yemen.
[3] Information Technology Department, University of Science and Technology, Ibb, Yemen.
*Corresponding author:author Email a.alshalabi@su.edu.ye and s.alhagree@su.edu.ye

ARTICLE INFO
Article history:
Received: January 16, 2023
Accepted: April 5, 2023
Published: April 30, 2023

Keywords
1. Arabic Sentiment Analysis
2. K-Nearest Neighbor
3. Levenshtein Distance
4. Google Play

ABSTRACT: The field of sentiment analysis of Arabic scripts remains a major challenge due to the features, characteristics, and complexity of the Arabic language. Few studies on Arabic Sentiment Analysis (ASA) have been conducted to the best of our knowledge when compared to English or other Latin languages. In addition, for ASA, very few studies have been conducted regarding the comments of the users (customers) of the apps available on the Google Play Store within various mobile application reviews. Most of the current studies have been conducted on datasets collected from Twitter user comments. In this paper, we propose a new approach to the analysis of sentiment in Arabic script based on the comments dataset of users of some mobile applications available on the Google Play Store. The proposed approach involves improving algorithms such as the Levenshtein distance (LD) algorithm for data preprocessing and then combining it with the K-Nearest Neighbor (K-NN) algorithm. Through the proposed approach, the results of the experiment were shown, investigating the impact of utilizing the K-NN and LD algorithms for ASA on mobile applications effectively. The experiments were carried out. The K-NN with LD algorithm has gained a better level of evaluation compared to the K-NN algorithm. K-NN with LD algorithm has achieved the highest accuracy, recall, precision, and F-score, which were 83.11% in accuracy, 66.30% in recall, 85.10% in precision, and 74.53% in f-score evaluation measure when =3.

CONTENTS

1. Introduction

It is a fact that the number of users of mobile applications is increasing considerably on a global scale. It is not only this, but also men could explain their ideas about events or vague issues via reviews using a variety of platforms like Google Play. By doing so, the number of content creators using Google Play steadily increases. Users post comments about the quality and content of the mentioned applications, as they are part of the content creators that use platforms like Google Play. It is commonplace to find government and private institutions exploiting users' opinions and expressions with regard to services or products through the Internet. Sentiment analysis and machine learning have an important role to play "towards the written text" to track users' behaviors or opinions regarding some applications. To address the issue of opinions, there has been much development of machine learning techniques to a certain extent for Arabic reviews, which are scripted in English [1]. Nevertheless, Arabic reviews constitute a significant domain for researchers to investigate ways in which machine learning and the analysis of sentiment can be developed to secure accurate and exact information automatically. A great number of users depend largely on reviews and ratings before they start downloading an app, particularly while similar options are available, which is significant for apps to improve their quality [1]. Concerning the number of Arabic scholarships in the Arabic language, it is dwindling. A novel approach is proposed in this paper for the analysis of Arabic sentiment, which relies on string matching via machine learning for automatic identification. This paper comes up with the following contributions:

- An LD algorithm based on Arabic sentiment classification using K-NN is developed.
- An approach for ASA based on string matching via K-NN is proposed. It can provide a platform for the performance evaluation of this research in Arabic string matching.
- Using the proposed approach, we can deal with the unique features and characteristics of the Arabic language and the range of misspellings.

This study aims at investigating the effect of utilizing the K-NN and LD algorithms for the analysis of Arabic sentiment on mobile applications. Moreover, the task of analyzing sentiment presumes its utility by using one of the following two methods: lexicon-based sentiment analysis (LBSA) or sentiment analysis of machine learning (MLSA) [2]. A vocabulary dictionary is used by LBSA for computing the polarity of the record of each text. For predicting text record polarity, MLSA uses machine learning (ML) models. Although MLSA proves to be more proficient, human-annotated data is needed for training on polarity prior to the prediction of the task [3]. The organization of the rest of the paper is as follows. Section 2 incorporates the major studies relevant to sentiment analysis based on app feedback data from Google. The methodology of this work is explained in Section 3. In Section 4, the findings and analysis of the proposed approach are incorporated. Finally, Section 5 concludes the main findings and offers some topics for further research.

2. Literature Review

To figure out the research gap in sentiment analysis, many important and relevant studies have been thoroughly studied and investigated. Sentiment analysis is the process of extracting useful patterns from textual data. These patterns of usage include categorizing and interpreting sentiment into positive, negative, or neutral comments from that data using techniques like machine learning. When using such techniques, the study users' information can be available on the web, including social networks and other platforms. Many papers are concerned with obtaining a solution for the analysis of sentiment based on Arabic string matching. In addition, the reviews based on sentiment analysis use various platforms, such as Google Play. The analysis of sentiment was applied to assist the Saudi government in developing the utility to control the breakout of COVID-19 [1]. Eight thousand reviews were gathered from the App Store and Google Play. Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree, and K-nearest neighborhood were applied. The findings revealed that K-Nearest Neighborhood offered the best possible result of 78.46%. In [2], a new dataset is constructed, which includes 51k user reviews. Approaches to feature engineering, including AFINN, Bing Liu lexicon, MPQA Subjectivity Lexicon, TF-IDF, BoWs, and the Google pre-trained Word2Vec, were integrated. Five machine learning models, including NB, Random Forest (RF), Logistic Regression (LR), Bagging, and SVM, were applied to the collected dataset. The highest accuracy they obtained was 93.17%, using the SVM. In [4], to analyze the sentiment of product reviews, they introduced a novel approach. Their proposed approach was based on the LD algorithm. Their proposed method is to read reviews across different websites for a specific product, which assists the user in selecting the product. In [5], to analyze the sentiment of Hinglish text, they introduced a novel approach. In their proposed approach, they used multiple data pre-processing algorithms, such as Soundex, LD algorithm, and stemming. Thereafter, they applied various classification models to identify the polarity of Hinglish text. The results of the experiment showed that the proposed approach was effective for sentiment analysis of Hinglish text. They used six machine learning models such as SVM, NB, LR, RF, and Decision Tree. Their results showed that SVM yielded the best accuracy of 70.90%, while NB, LR, RF, and Decision Tree yielded an accuracy of 59.09%, 70%, 66.36%, and 64.54%, respectively. In [6], for the Arabic sentiment analysis of text data representation, they introduced a new method. Their proposed method was based on the LD algorithm. By adding features computed using the LD algorithm and Bag of Words. The data set used was Arabic comments in the Moroccan dialect, in order to discover the polarity. They applied their proposed method. They used the deep neural network classifier and got an accuracy result of 62%. In [7], for a sentiment analysis of the school zoning system on YouTube, they introduced a new method. Their proposed method was based on the LD algorithm and the K-NN algorithm. They combined the two algorithms in order to improve the accuracy of the sentiment analysis of the school zoning system on YouTube. The results of the experiment showed that the proposed approach is effective for sentiment analysis of school zoning system on YouTube. Their results showed that combining the K-NN and LD algorithms yielded the best accuracy of 65.625% when a value of k = 3. The K-NN algorithm obtained an accuracy of 50% for the values k = 7 and k = 3. In [8], they used sentiment analysis. They are interested in analyzing Algerian reviews on the application store. They created a dataset containing 50,000 reviews (comments) in languages (Algerian dialects, French, and Arabic) from users of some applications available on Google Play. Which it characterized by the specificity of writing them using different languages (Algerian dialects, French, and Arabic), which makes them difficult to process. They used two methods to analyze these reviews in languages (Algerian dialects, French, and Arabic): the automatic approach was based on machine learning and the lexical-based approach. In addition, they used the LD algorithm to compare review words to vocabulary. They used six machine-learning models such as SVM, NB, K-NN, ANN, and DT. Their results showed that SVM yielded the best accuracy of 75 %, while NB, K-NN, ANN, and DT yielded an accuracy of 62%, 61%, 70%, and 62%, respectively. In [9], they created a dataset containing 51,000 Arabic reviews (comments) from users of some applications available on Google Play. These applications provide government services. User comments were extracted using the scraping technique. Moreover, they extracted user reviews from six applications on Google Play. In addition, they discussed the future goals of the dataset for improvement tasks and software development such as sentiment analysis. In [10], they investigated the impact of applying different spelling correction algorithms, such as Peter Norvig's algorithm and the LD algorithm for ASA. Before carrying out sentiment analysis, they corrected the spellings of the unknown words found on some social media sites, such as Twitter. For sentiment analysis, they evaluated spelling correction algorithms by comparing the polarities obtained from a sentiment analysis algorithm with the polarization of manually annotated tweets. With the Levenshtein distance-based algorithm, they got improvements in terms of the percentage of matched polarity, which was 1.6%. With Peter Norvig's algorithm, they got improvements in terms of the percentage of matched polarity, which was 2.0%. In [11], the aim was that customers get satisfaction in terms of digital banking in Indonesia using SA from Twitter. Data were gathered from three digital banks in Indonesia, specifically Blu, Jago, and Jenius. A total of 34,605 tweets were collected and analyzed. SA was presented using nine standalone classifiers: NB, K-NN, Decision Tree, LR, SVM, RF, Adaptive Boosting, Light Gradient Boosting Machine, and eXtreme Gradient Boosting. Two approaches are used: soft voting and hard voting. The findings reveal that SVM achieves the best performance in comparison with other standalone classifiers, with an F1 score of 73.34%. The ensemble approach outperformed using a stand-alone classifier, and soft voting with the 5-best classifiers presented the best results, with an F1 score of 74.89%. In [12], for a sentiment analysis of the school zonation system policy on reviews based on YouTube and Facebook, they introduced a new method. Their proposed method was based on the LD algorithm and the K-Means algorithm. They combined the two algorithms in order to improve the accuracy of the sentiment analysis of the school zonation system policy on reviews based on YouTube and Facebook. The results of the experiments showed that the proposed approach is effective for sentiment analysis of the school zonation system policy on reviews based on YouTube and Facebook. Their results showed that combining the LD algorithm and the K-Means algorithm yielded the best accuracy of 90%, while the K-Means algorithm obtained an accuracy of 84%. In [13], they introduced a new method for Indonesian sentiment analysis based on the public comments about the Covid-19 vaccine data in that language. They created a dataset containing 2394 comments from users of the Indonesian Ministry of Health's Instagram account. For the preprocessing of the data, they used the LD algorithm. Their results showed high accuracy by combining the LD algorithm with the NB algorithm. Their results showed that combining the LD and NB algorithms yielded the best accuracy of 71%, while the NB algorithm obtained an accuracy of 61%. In [18-19], they introduced a new method to ASA based on the mobile app comments data in Arabic. For the preprocessing of the data, they used the LD algorithm. Their results showed high accuracy by using the LD algorithm with the NB algorithm. Their results showed that combining the LD and NB algorithms yielded the best accuracy of 96.40% when a value of k = 9, while the NB algorithm obtained an accuracy of 95.80% for the values k = 9. In addition, all these studies are, to the best of our knowledge, related to Arabic sentiment analysis. There is a gap. In [21], they introduced a new method that is reliable and easy to use, NB. The C4.5 method is also very popular for solving the decision tree problem, which they used for the sentiment classification process. They used the LD method to compare two strings for the word normalization process. The method flow started with text preprocessing the dataset with LD. They split the dataset into two for the C4.5 and NB classification processes. Their results showed that the decision tree and NB algorithms yielded the best accuracy of 85.6%, while combining the LD, NB, and decision tree algorithms yielded an accuracy of 85.3%. The difference is 0.3%. Making the LD algorithm does not significantly affect the classification results. In [22], the purpose is to find out the Arabic dialects on social platforms in an unsupervised manner. To do so, they used an Algerian dialect lexicon that is based on the Algerian dialect. They also suggested an approach based on an algorithm that presents three kinds of identification: 1) total (when the term is identified), 2) partial (when the term is partially identified with suffixes and prefixes), and by using the improved LD algorithm (based on the classical LD algorithm) considering the number of characters of the words being compared. In addition, in their research, they used their algorithm on a corpus of 100 messages that were gathered using the Facebook API. They got a rate exceeding 60%. In fact, there is a huge number of studies relevant to sentiment analysis based on the LD algorithm. The most recent studies are summarized in Table 1.

3. Methodology

This paper adopts a novel approach for developing sentiment analysis of Arabic text using the K-NN algorithm based on the LD algorithm [5] [7] [13] [18]. This methodology is used to perform Arabic sentiment analysis on mobile application reviews. The proposed approach works in three phases, as shown in Fig. 1.

Table 1. Summaries of the studies on sentiment analysis based on the LD algorithm

References	Year	Study title	Technology	Result
[18]	2022	Arabic sentiment analysis (ASA) on mobile applications using the LD and NB algorithms	LD algorithm	Accuracy 96.40%
[13]	2022	Sentiment Analysis of the Covid-19 Vaccine using the NB Algorithm and LD Word Correction	algorithm	Accuracy 71%
[4]	2020	A novel sentiment classification of product reviews using LD	LD algorithm	---------------
[5]	2020	A novel approach for sentiment analysis of Hinglish text	LD algorithm	Accuracy 70.90%
[6]	2018	Arabic sentiment analysis using an LD-based representation approach	LD algorithm	Accuracy 62%
[7]	2019	Sentiment analysis of the school zoning system on YouTube social media using the K-NN with LD algorithm	LD algorithm	Accuracy 62.625%
[8]	2020	Sentiment analysis in the Google Play Store: Algerian reviews	LD algorithm	Accuracy 75%
[12]	2021	K-means algorithm and LD algorithm for sentiment analysis of the school zonation system policy.	LD algorithm	Accuracy 90%
[21]	2020	Mobile application review sentiment analysis using the combined NB and C4.5 algorithms based on LD word normalization	LD algorithm	Accuracy 85.3%
[10]	2018	Improving Topical Social Media Sentiment Analysis by Correcting Unknown Words Automatically	LD algorithm	Got improvements 1.6 %
[22]	2016	Arabic Dialect Identification with Unsupervised Learning (Based on a Lexicon) Application Case: Algerian Dialect	LD algorithm	Accuracy 60%
[23]	2017	SENTIMENT ANALYSIS OF THE NEWS DATA BASED ON SOCIAL MEDIA	LD algorithm	------------
[24]	2013	SENTIMENT CLASSIFICATION OF MOVIE REVIEWS USING LD	LD algorithm	Accuracy 80%

3.1 Data Preparation of Dataset

The original dataset consists of 51,767 user comment reviews from mobile applications [1] [9]. All user review comments are written in Arabic. For a test of the method proposed in this paper, we extracted 8566 user comments to construct a dataset as follows: Training data is part of the dataset that we train to measure the accuracy or performance of the proposed method.

The training data used comes from mobile application (app) reviews. The dataset was built on many of the 8566 records of user reviews, which are labeled as negative, neutral, or positive, and then saved in CSV format. Table 2 shows a sample of reviews that contain the ID number, comment text, and polarity. Figure 2 shows the distribution of polarity scores in the training dataset, which contains 2121 instances that were labeled as negative comments, 1000 instances as neutral comments, and 5445 instances as positive comments.

3.2 Method of Data Preprocessing

An important step that helps enhance and extract meaningful insights from data is data pre-processing:

Fig. 1. The Model proposed for Arabic sentiment polarity classification.

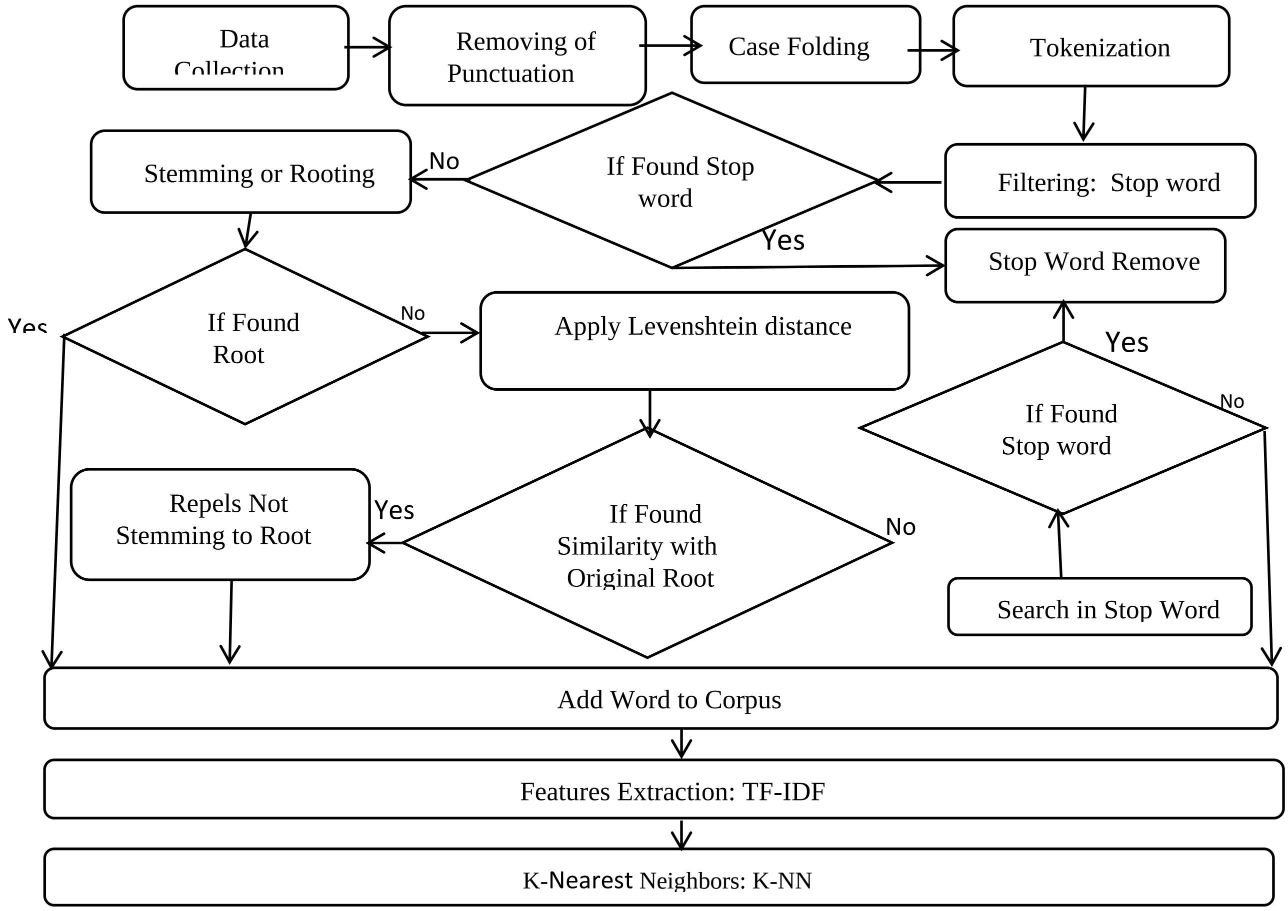

Table 1. A sample of reviews in the training data.

ID	Sample of review	Polarity
8492	افضل من التطبيق السابق بكثير أبهجني ما فعلتم اتمنى لكم مزيدا من النجاح	Positive
6900	ممتاز البرنامج ماشاءالله تبارك الرحمن.. شكرا على الخدمة	Positive
1583	سجلت قبل فترة ولم تأتي الموافقة	Neutral
6648	البرنامج عطططططططلان مايقبل النفاذ الوطني للاسف الشديد	Negative
8546	لا يعمل... حاولت التسجيل تظهز رسالة لا يمكن تسجيلك في النسخة التجريبية	Negative
6057	تجربة سيئة وتطبيق غبي لم أستطيع عمل التسجيل الجديد	Negative
6228	برنااااامج جداااا رائع.. وصلني الكود وسهل الاستخدام	Positive

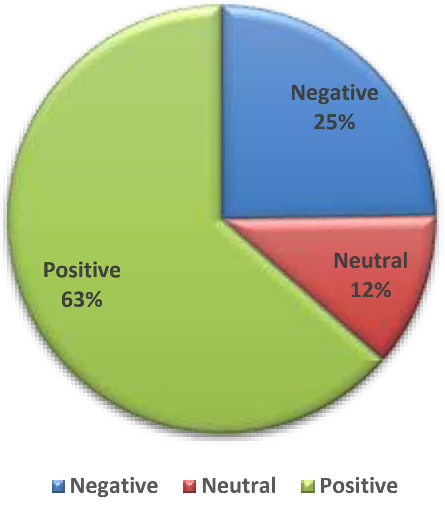

Fig. 2. Distribution of polarity scores for the training dataset

Table 3. A sample of reviews for the removal of data.

No	Sample of review	Removing
789	إلى الامام ياوطني الحبيب 100%	100%
2604	لايمكن تسجيل تاريخ الميلاد 1935 !!	1935 !!
2878	لايدعم الاجهزه الجديدة من هواوي p40	p40
3016	أنصح بتنزيل هذا التطبيق ويستحق 5نجوم	5

The data preprocessing method helps remove the inconsistencies and errors present in the data. Sometimes it is the inconsistencies of the data that create missing or illogical important information that affects the accuracy of the data. The following actions are steps in data pre-processing methods that are summarized as follows:

(1) Removing (2) Folding of Case (3) Tokenization (4) Stop Word: Filtering (5) Rooting or Stemming (6) LD algorithm. This approach is used for removing stop words and finding close relationships between words in the Arabic dictionary, e.g., the word **"ممتاااازة"** with **"ممتاز"** and the word **"برنااااامج"** with **"برنامج"** and the word **"عططططططططلان"** with **"عطلان"** and the word **"جدااااا"** with **"جدا"** have a close relationship, i.e., they sound similar and have an exact meaning. Data pre-processing involves the following details:

3.2.1 Folding of case

In this second stage, data was collected from the uniform cases or letters contained in each comment from mobile applications (apps). Uniformizing letters was done from "إ ,أ ,آ "letters converted to "ا" letters and "ة,ي" letters converted to "ه,ى" letters. An example of replacing initial إ ,أ ,آ with ا.

3.1.1 Tokenizing

Tokenizing (splitting a text) is regarded as the third stage that is identified as a process of separating or cutting the input string based on each constituent word. Concerning the case-folding process for tokenization, it is marked by (-) as a list of word compilers.Tokenization indicates splitting each string into a single word/token. In this process, each sentence/string is divided into several segments like phrases, keywords, symbols, and words. These segments are named tokens. Thus, tags and punctuation marks are also abandoned. Furthermore, the letters are altered to lowercase. Table 4 shows an example of tokenizing a sample of reviews.

3.1.2 Stop word: Filtering

The fourth stage is filtering. Many terms (words) that are not important in identifying the polarity of the document are called Stop Words. Improving the performance of sentiment analysis is done by reducing the size of the vector and eliminating these words. In addition, filtering is the process of removing any unnecessary data from the sentence and stop words. In this paper, the method of Khoja is used as it supports the Arabic language. Table 5 shows a sample of reviews for the stop word of the data. Table 6 shows sample reviews for the stop word, which is not detected by Khoja, and the core of our work in this paper is to improve the LD algorithm in order to delete these words, as shown in Table 6. Details are provided in the subsection.

3.1.3 Rooting: Stemming

The last stage is stemming, in the use of words in a sentence, where there are things that are not recognized by the rules or spelling dictionaries. Stemming is a common task in sentiment analysis. We used the full stemmer in this work to improve the performance of sentiment analysis by reducing the size of the vector and finding the root of these words. Thus, the word vector size will be reduced considerably. This work has employed the Arabic stemmer offered by Khoja. Table 7 shows a sample of reviews for the stemming. Table 8 shows sample reviews for stemming, which is not detected by Khoja. The core of our work in this paper is to improve the LD algorithm in order to detect these words, as shown in Table 8. Details are provided in this subsection. Thus, if a similar word is not found, the LD algorithm [7] is applied to get it from the root.

Table 5. A sample of reviews for the stop word of data

No	Stop word	No	Stop word
1	أما	6	اول
2	الذين	7	به
3	لا	8	ولذلك
4	تحت	9	من
5	عليكم	10	في

Table 6. A sample of reviews for the stop word of data difficult to detect

No	Stop word	No	Stop word
1	أاااااما	6	اااااااول
2	اااااااااااااالذين	7	بببببببببببببه
3	لالالالا	8	وووووولذلك
4	تحتتتتتتت	9	منننننننن
5	عليييييكم	10	فففففففففففففي

Table 7. A sample review for stemming

No	Original word	Stemming	No	Original word	Stemming
1	انضباط	ضبط	6	بسيط	بسط
2	رائع	روع	7	الأصلي	صلي
3	جبار	جبر	8	ابداع	بدع
4	يحمله	حمل	9	خصوصا	خصص
5	عملي	عمل	10	ناجح	نجح

Table 8. Sample reviews for not stemming

No	Original word	No	Original word
1	عيالادات	6	خدووووووم
2	ماقصرتو	7	برناااامج
3	مايشتغل	8	ررررررووعههhe
4	الميلاد	9	ممتااازة
5	بلغلط	10	عططططططلان

3.1.4 Levenshtein distance algorithm

In the Levenshtein distance (LD) algorithm, given two names X and Y represented as strings of n and m characters, respectively, the LD, aka, refers to the minimum cost of editing operations (inserting, deleting, and substituting) to convert X to Y [14], [15], [25]. For example, if X=" Zantac‖" and Y=" Xanax‖", the edit distance is 3, as the minimum transformation implies two substitution operations ("Z" → "X" and "c" → "x") and one deleting operation (letter "t"). In this work, editing, inserting, and deleting operations have a cost of 1 and 1, respectively. Specifically, the edit distance between s and t is given by Lev (i, j), which is computed using the recurrence formula in Eq. (1).

$$\mathrm{Lev}_{s,t}(i,j) = \begin{cases} \mathrm{Max}(i,j) & (i = 0 \text{ or } j = 0) \\ \mathrm{Min}\begin{cases} \mathrm{Lev}_{s,t}(i, j-1) + 1 \text{ , is 1.} \\ \mathrm{Lev}_{s,t}(i-1, j) + 1 \text{ , is 1.} \\ \mathrm{Lev}_{s,t}(i-1, j-1) + w_1 \quad xi \neq yj \end{cases} \end{cases} \quad (1)$$

This equation computes the replacement cost (substitution) that takes a real value in the interval [0, 1]. Its value is one when the source does not equal the target, and it is set to zero otherwise. The LD algorithm is a well-established mathematical algorithm for measuring the edit distance between words and can specifically weight deletions and insertions in this paper. After preprocessing the data, it is expected that the data will become more uniform. Table 9 shows the dataset statistics preprocessed. It was previously discussed that words are deleted and their roots are found through Stop Word and stemming, respectively. Moreover, there are many words that are not Stop Words and also do not have a root as a result of typos, spelling errors, and keyboard errors, especially when using the mobile phone keyboard. Therefore, we suggested applying the LD algorithm in order to find similar words and search for the root of the word similar to it with a specified threshold and Table 10. He shows us some examples. In addition, there are words for which the root was not discovered when using the LD algorithm, although these words are correct and have roots. They were, however, written by the user using the excitation method. Therefore, in this paper, we improved the LD algorithm in order to discover these words and find their roots. Fig. 3 shows the enhancement of the LD algorithm by adjusting the weights of the adding and deleting processes, where this type of error is tolerated by repeating the letter consecutively within the word.

Table 9. Statistics of the preprocessed dataset

No	Unique word	22887
1	Punctuation	42
2	Not Letter	244
3	Stopword	889
4	Stemming word	18095
5	Not Stemming	3617

Table 10 shows words before and after handling the errors of deleting and inserting. As shown in figure 3, the function of inserting and deleting operations computes the weights of the different cost components. After that, comments' records were obtained, like in Table 10, where we find the comments being presented prior to processing while remaining in their original shape, "Original comments" Sample review" and the comments after processing using the LD algorithm.

The compatibility of edit operations in the proposed algorithm has been elaborated in the following examples and shown in Figure 4. Figure 4 shows how the proposed algorithm measures the distance between the source(s) "**عياادات**" and target(t) "**عيادات**" [25][26]. The distance is 1. Therefore, the similarity between them is 85%. It is obvious that the proposed algorithm gives a very similar result for inserting "ا " because it is a repeated letter. It is obviously the proposed algorithm that inserts 'عياادات' from t into s.

Table 10. A sample of review not stemming using the LD algorithm

No	Not stemmed	Original word	Root	Distance	Similarity
1	عياادات	عيادات	عود	1	0.857
2	ماقصرتو	ماقصرتوا	قصر	1	0.875
3	مايشتغل	لايشتغل	شغل	1	0.857
4	الميلاد	المبلاد	بلد	1	0.857
5	بلغلط	بالغلط	غلط	1	0.833
6	ممتاااازة	ممتاز	ميز	1	0.833

```
Function. To calculate the sum of the Arabic deletion and insertion.
Input: Two Letters (Letter)
Output: Cost of Inserting and Deleting Distance (cost)
Decimal Inserting and Deleting Op (i,j)
 If (sor_name[i] == sor_name [i-1])
   csosttID=0.1;
elseIf (tar_name [j] == tar_name [j-1]
    csosttID=0.1;
Return csosttID
```

Fig 3. The function of deletion and insertion operations

		ع	ي	ا	ا	د	ا	ت
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
ع	**1**	0	1	2	3	4	5	6
ي	**2**	1	0	1	2	3	4	5
ا	**3**	2	1	0	0	3	4	5
د	**4**	3	2	2	1	0	1	2
ا	**5**	4	3	3	1	1	1	2
ت	**6**	5	4	4	2	2	3	1

Fig 4. The Distance Between (S) “عياادات” and (t) “عيادات” in the LD Algorithm

3.2 Features Extraction: TF-IDF

It is unanimous that four sample documents are applied in this research paper, which are the results of the steaming stage: the data preprocessing method, the LD method, the TF-IDF method, and the K-NN method [16] [17] [20]. TF-IDF method: term frequency-inverse document frequency. The sample data used in this paper above is computed using the TF-IDF formula, which aims to classify the text of a document. A strength of our approach is mitigating the number of singletons in root and stop words to improve the performance of ASA by reducing the size of the vector, finding the root of these words, and eliminating stop words. Thus, the word vector size will be reduced considerably. Table 11 shows statistics of vector size before and after the proposed method. This paper has numerous research objectives that must be completed to accomplish the research goals. As we proceed, the research goals are explained clearly. Some classification algorithms, such as K-NN with k=2, k=3, k=4, and k=5 classifiers, were identified by comparing their accuracy to machine learning techniques. After the verification of the ongoing approach: evaluation of the results of experiments and performance of the algorithm with the used approach via measures like accuracy. In this paper, machine-learning models are adapted and applied, including machine learning as in K-NN when k=2, k=3, k=4, k=5, with the sigmoid equation for the analysis of Arabic sentiment.

3.3 K-Nearest Neighbors

Given a dataset with X as the input matrix and Y as the label variable, K-NN is a classification method that estimates the conditional distribution of Y given X and assigns an input instance to the class label with the highest probability [20]. For a positive integer k, K-NN finds k observations closest to an input instance x_0 and estimates the conditional probability that it belongs to class j using the following equation:

$$\Pr(Y = j|X = x_0) = \frac{\sum_{i\in N_0} I(y_i = j)}{k} \qquad (2)$$

Where N_0 is the set of k-nearest instances and I(yi=j) is an indicator variable that equals 1 if (xi,yi) belongs to class j and 0 otherwise. After calculating the probabilities, K-NN classifies the input instance x_0 into the class with the highest probability. The first step in K-NN is calculating the distances between the input instance and the reference (labeled) instances in the training data. To do so, a distance metric is needed. One of the most commonly used metrics is the Euclidean distance, which is calculated below.

$$d(u,v) = \sqrt{\sum_{i=1}^{p} (u_i - v_i)^2} \qquad (3)$$

where $u = (u_1, u_2, \ldots, u_p)$ and $v = (v_1, v_2, \ldots, v_p)$.

Table 11. The statistics of vector size

No	Attribute	Before the proposed method	After the proposed method
1	Columns	3447	2791
2	Rows	Fixed	Fixed

4 Experiments and Results

In this section, we investigate the impact of utilizing the K-NN and LD algorithms for ASA on mobile applications through the experiments that have been carried out in this paper.

4.1 Evaluation criterion

Four evaluation metrics were utilized in this paper to evaluate the adopted approach (LD algorithm with K-NN algorithm). They are accuracy, precision, recall, and f-score measures. Accuracy indicates the ratio of the amount of accurately estimated samples to the total sum of predicted samples.

Where:

- **Accuracy**: Accuracies for classification tasks, as in Eq. (4).

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \; (4)$$

- **Recall:** recall is computed using Eq. (5).

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (5)$$

- **Precision**: precision is computed using Eq. (6).

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \; (6)$$

- **F-score:** F-score is computed using Eq. (7).

$$\text{F} - \text{score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \; (7)$$

Where TP is the true positive, FP the false positive, TN the true negative, and FN the false negative.

4.2 Experimental results

The researcher carried out the experiment based on the accuracy value, as shown in Table 12. The discussion of the experiment as a means to validate the adopted approach with the value of K. The adopted approach reveals that many uses over the compared algorithm. Thus, the chosen algorithm yields more exact results than the other algorithm, which is manifested in Table 12 and Fig. 4. Therefore, while reviewing mobile applications from the Google Play dataset, K-NN with LD algorithm when k=3, classifier performs better (accuracy—83.11%, precision—85.10%, recall—66.30% and F1-score—74.53%) compared with that of other K-NN when k=3, performs (accuracy—82.78%, precision—85.13%, recall—65.93%, and F1-score—74.31%). Fig. 5 delineates graphically the average values of classification accuracy, recall, precision, and F1-score of K-NN and K-NN with LD when k=2, k=3, k=4, k=5. Accuracy results from the K-NN algorithm, namely in the sum of 82.78% to the value k = 3. However, the accuracy of the combination of K-NN with the LD algorithm of 83.11% with a value of k = 3 applies better. Moreover, our modifications drastically improved accuracy.

Table 12. The results of ASA using the LD algorithm with the K-NN algorithm

K Value	Techniques	Accuracy	Recall	Precision	F-score
K=2	K-NN	81.36%	66.53%	82.44%	73.64%
	K-NN with LD	82.00%	67.52%	83.08%	74.50%
K=3	K-NN	82.78%	65.93%	85.13%	74.31%
	K-NN with LD	**83.11%**	**66.30%**	**85.10%**	**74.53%**
K=4	K-NN	79.65%	61.35%	80.13%	69.49%
	K-NN with LD	79.98%	61.46%	79.51%	69.33%
K=5	K-NN	78.87%	59.02%	80.46%	68.09%
	K-NN with LD	79.50%	59.88%	80.21%	68.57%

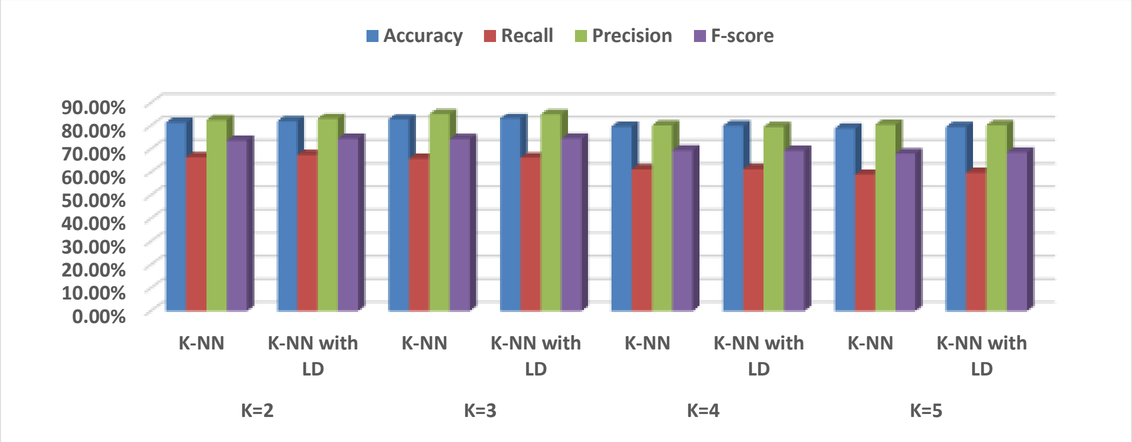

Fig. 5. Graphical representation of performance metrics in the LD algorithm with the K-NN algorithm

5 Conclusion

This paper presented an approach new to performing Arabic sentiment analysis on the Google Play Store's applications. The combination of the Levenshtein distance algorithm with K-NN classified the polarity of the Arabic sentiment analysis. A strength of our approach is mitigating the number of singletons in root and stop words to improve the performance of Arabic sentiment analysis by reducing the size of the vector, finding the root of these words, and eliminating stop words. Thus, the word vector size will be reduced considerably. The proposed approach K-NN with LD when k=3, classifier performs better (accuracy—83.11%, precision—85.10%, recall—66.30%, and F1-score—74.53%) in comparison with that of other K-NN when k=3, performs (accuracy—82.78%, precision—85.13%, recall—65.93%, and F1-score—74.31%). For future work, incorporating the Levenshtein distance algorithm and lexicon-based approach and prolonging the amount of data (Big Data).

6 References

[1] Hadwan, M., Al-Haery, M., Al-Sarem, M., & Saeed, F. “Arabic Sentiment Analysis of Users’ Opinions of Governmental Mobile Applications. Computers”, Materials and Continua, 72(3), 4675-4689, **2022**. https://doi.org/10.32604/cmc.2022.027311

[2] Fahd Alqasemi, Amira Abdelwahab, and Hatem Abdelkader. “Constructing automatic domain-specific sentiment lexicon using KNN search via terms discrimination vectors”. International Journal of Computers and Applications 41.2: 129-139 ,**2019**. https://doi.org/10.1080/1206212X.2017.1409477

[3] Alqasemi, F., Salah, A. H., Abdu, N. A. A., Al-Helali, B., & AlGaphari, G. “Arabic Poetry Meter Categorization Using Machine Learning Based on Customized Feature Extraction”. International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE) (pp. 1-4). IEEE, **2021**. https://doi.org/10.1109/ITSS-IoE53029.2021.9615302

[4] Anjali, T.; Krishnaprasad, T. R.; Jayakumar, P., “A Novel Sentiment Classification of Product Reviews using Levenshtein Distance”. In: 2020 International Conference on Communication and Signal Processing (ICCSP). IEEE. p. 0507-0511, **2020**. https://doi.org/10.1109/ICCSP48568.2020.9182198

[5] Rao, Himanshu Singh, Jagdish Chandra Menaria, and Satyendra Singh Chouhan. “A Novel Approach for Sentiment Analysis of Hinglish Text.” Mathematical Modeling, Computational Intelligence Techniques and Renewable Energy: Proceedings of the First International Conference, MMCITRE 2020. Springer Singapore, **2020**. https://doi.org/10.1007/978-981-15-9953-8_20

[6] Essatouti, B., Khamar, H., El Fkihi, S., Faizi, R., & Thami, R. O. H. . “Arabic sentiment analysis using a levenshtein distance based representation approach.” In 2018 IEEE 5th International Congress on Information Science and Technology (CiSt) (pp. 270-273). IEEE ,**2018**. https://doi.org/10.1109/CIST.2018.8596379

[7] Anggraini, N., & Tursina, M. J. “Sentiment analysis of school zoning system on Youtube

social media using the K-nearest neighbor with levenshtein distance algorithm". In 2019 7th International Conference on Cyber and IT Service Management (CITSM) (Vol. 7, pp. 1-4). IEEE ,**2019**. https://doi.org/10.1109/CITSM47753.2019.8965407

[8] Chader, A., Hamdad, L., & Belkhiri, A.. "Sentiment analysis in google play store: Algerian reviews case". In International Symposium on Modelling and Implementation of Complex Systems (pp. 107-121). Springer, Cham ,**2020**. https://doi.org/10.1007/978-3-030-58861-8_8

[9] Al-Shamani, M., Al-Sarem, M., Saeed, F., & Almutairi, W.. "Designing an Arabic Google Play Store User Review Dataset for Detecting App Requirement Issues. " Advances on Smart and Soft Computing. Springer, Singapore, 133-143 ,**2022**. https://doi.org/ 10.1007/978-981-16-5559-3_12

[10] Alfred, R., & Teoh, R. W. "Improving topical social media sentiment analysis by correcting unknown words automatically". In International Conference on Soft Computing in Data Science (pp. 299-308). Springer, Singapore, **2018**. https://doi.org/10.1007/978-981-13-3441-2_23

[11] Bramanthyo Andrian, Tiarma Simanungkalit, Indra Budi, Alfan Farizki Wicaksono, "Sentiment Analysis on Customer Satisfaction of Digital Banking in Indonesia" , (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 13, No. 3, **2022**. https://doi.org/10.14569/IJACSA.2022.0130356

[12] Al Farisi, M. H., Wardhani, L. K., Matin, I. M. M., Durachman, Y., Adelina, R., & Nurdin, F. "K-Means Algorithm and Levenshtein Distance Algorithm for Sentiment Analysis of School Zonation System Policy. "2021 Sixth International Conference on Informatics and Computing (ICIC). IEEE, 202), **2021**. https://doi.org/10.1109/ICIC54025.2021.9632943

[13] Prasastio, F. R., & Heriyanto, W. K. "Sentiment Analysis of the Covid-19 Vaccine Using the Naive Bayes Algorithm and Levenshtein Distance Word Correction",**2022**. https://doi.org/10.31515/telematika.v19i1.6577

[14] Al-Hagree, S., Abdulmalik, S., Alsurori, M., & Al-Sanabani, M. "An Enhanced Algorithm for Matching Arabic Names Entered by Mobile Phones". In 2019 First International Conference of Intelligent Computing and Engineering (ICOICE) (pp. 1-8). IEEE ,**2019**. https://doi.org/10.1109/ICOICE48418.2019.9035148

[15] Abdulmalek, S., Salah, A. H., Alsurori, M., Hadwan, M., Aqlan, A., & Alqasemi,. "Levenstein's Algorithm on English and Arabic: A Survey". In 2021 International Conference of Technology, Science and Administration (ICTSA) (pp. 1-6) ,**2021**. https://doi.org/10.1109/ICTSA52017.2021.9406547

[16] Pravina, Arsya Monica. "Sentiment Analysis of Delivery Service Opinions on Twitter Documents using K-Nearest Neighbor. " JATISI (Jurnal Teknik Informatika dan Sistem Informasi) 9.2: 996-1012,**2022**. https://doi.org/10.35957/jatisi.v9i2.1899

[17] Naeem, M. Z., Rustam, F., Mehmood, A., Ashraf, I., & Choi, G. S. "Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning algorithms". PeerJ Computer Science, 8, e914,,**2022**. https://doi.org/10.7717/peerj-cs.914

[18] Al-Hagree, S., & Al-Gaphari, G. Arabic Sentiment Analysis on Mobile Applications Using Levenshtein Distance Algorithm and Naive Bayes". In 2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA) (pp. 1-6). IEEE ,**2022**. https://doi.org/10.1109/eSmarTA56775.2022.9935492

[19] Al-Hagree, S., & Al-Gaphari, G. "Arabic Sentiment Analysis Based Machine Learning for Measuring User Satisfaction with Banking Services' Mobile Applications: Comparative Study". In 2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA) (pp. 1-4). IEEE ,**2022**. https://doi.org/10.1109/eSmarTA56775.2022.9935486

[20] Al-Helali, Baligh, "A new imputation method based on genetic programming and weighted KNN for symbolic regression with incomplete data. " Soft Computing 25.8: 5993-6012, **2021**. https://doi.org/10.1007/s00500-021-05590-y

[21] arrofi reza satria , sigit adinugroho , suprapto, "Mobile Application Review Sentiment Analysis using the Combined Naïve Bayes and C4.5 Algorithm based on Levenshtein Distance Word Normalization, urnal Pengembangan teknologi informasi dan ilmu komputer, Vol. 4, No. 11, November, hlm. 4154-4163, **2020**.

[22] Guellil, I., & Azouaou, F. "Arabic dialect identification with an unsupervised learning (based on a lexicon). Application case: Algerian dialect". In 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and

Applications for Business Engineering (DCABES) (pp. 724-731). IEEE, **2016**. https://doi.org/10.1109/CSE-EUC-DCABES.2016.268

[23] Shahare, F. F. "Sentiment analysis for the news data based on the social media". In 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1365-1370). IEEE, **2017**. https://doi.org/10.1109/ICCONS.2017.8250692

[24] Ahn, K. M., Kim, Y. S., Kim, Y. H., & Seo, Y. H. "Sentiment classification of movie reviews using Levenshtein distance". Journal of Digital Contents Society, 14(4), 581-587, **2013**. https://doi.org/10.9728/dcs.2013.14.4.581

[25] Al-Hagree, S., Al-Sanabani, M., Alalayah, K. M., & Hadwan, M. Designing an accurate and efficient algorithm for matching Arabic names. In 2019 First International Conference of Intelligent Computing and Engineering (ICOICE) (pp. 1-12). IEEE, **2019**. https://doi.org/10.1109/ICOICE48418.2019.9035184

[26] Al-Sanabani M, Al-Hagree S. Improved an algorithm for Arabic name matching. Open Transactions on Information Processing 2374–3778, **2015**.