



Rule-Based Arabic Sentiment Analysis using Binary Equilibrium Optimization Algorithm

Hichem Rahab¹ · Hichem Haouassi¹ · Abdelkader Laouid²

Received: 24 March 2022 / Accepted: 22 August 2022
© King Fahd University of Petroleum & Minerals 2022

Abstract

With the development of websites and social networks, Internet users generate a massive amount of comments and information on the Web. Sentiment analysis, also called opinion mining, offers an opportunity to mine the people's sentiments and emotions from the textual comments. In the last decade, sentiment analysis has been applied in research areas such as recommendation and support systems and has become an area of interest for many researchers. Therefore, many studies have been carried out on English, while other languages, such as Arabic, received less attention. Increasingly, sentiment analysis researchers use machine learning due to its excellent performance. However, the generated models are black boxes and non-interpretable by the users. The rule-based classification is a promising approach for generating interpretable models. This work proposes a classification rule-based Arabic sentiment analysis approach together with a new binary equilibrium optimization metaheuristic algorithm as an optimization method for classification rule generation from Arabic documents. The proposed approach has been experimented on the Opinion Corpus for Arabic (OCA) and generates a classification model of thirteen rules. The comparison results with state-of-the-art methods show that the proposed approach outperforms all other white-box models regarding classification accuracy.

Keywords Sentiment analysis · Rule-based classification · Arabic · Natural language processing · Associative classification · Equilibrium optimization algorithm

1 Introduction

Today's advancements in online communities enable the widespread of user-generated content. Such content allows a large community of simple users to generate a considerable mass of information that needs powerful tools to mine it [1]. The importance of user-generated content is dual; it allows organizations to have feedback from clients about their products and services. It also provides customers with

reviews posted by other users and their summarizations. Thus, one's view of the world is primarily influenced by others' views [2].

Sentiment analysis is the most crucial field in natural language processing (NLP), where the goal is to mine people's opinions about events, products, etc. The main task of sentiment analysis, also known as opinion mining [3], is the classification of opinionated documents into positive, negative, and neutral classes [4]; also, more classifications are possible [5–7].

Arabic is one of the ten top languages used on the Internet by 2021.¹ There are three forms of Arabic: classical Arabic, modern standard Arabic, and Arabic. In addition to these official forms, Arabic users on the Web use Arabizi [8–11] or Romanized Arabic [12, 13] as a form of Arabic text writing with Latin script [14]. The discretization is an additional dimension of Arabic language complexity when diacritics (light vowels) are essential to determine the sense of a word

✉ Hichem Rahab
rahab.hichem@univ-khenchela.dz

Hichem Haouassi
haouassi.hichem@univ-khenchela.dz

Abdelkader Laouid
abdelkader-laouid@univ-eloued.dz

¹ ICOSI Laboratory, Abbes Laghrour University, BP 1252, 40004 El Houria, Khenchela, Algeria

² LIAP Laboratory, El Oued University, PO Box 789, 39000 El Oued, Algeria

¹ <https://www.internetworldstats.com/stats7.htm>



or its part of speech category. These signs are omitted in Arabic texts in almost all cases [15–17].

Compared to English sentiment analysis, Arabic sentiment analysis (ASA) branch is still in its infancy [18]. We found in the literature several used approaches to resolve the problem of ASA. However, one can group the available approaches into two main categories [19]: machine learning or corpus-based [20–22] and lexicon-based approaches [23–27].

In the literature, many works adopt the machine learning approach due to its simplicity and the large available annotated corpora [28–30]. Machine learning models proposed in the literature are, in most cases, black-box methods when they do not help in producing human-interpretable sentiment analysis models [2]. Thus, despite considerable performance achieved using traditional black-box machine learning classifiers, they give no information about why a document is assigned to one class or another [1].

In contrast, white-box approaches use classification rules (CRs) to generate interpretable models [31]. In almost all rule-based approaches, especially lexicon-based approaches, the CRs are generated manually by decision-making managers [1]. However, given the large size of the corpora in general, it is impossible to do this manually. In this paper, we address the automatic generation of CR from the OCA corpus. The CR generation is considered a combinatorial optimization problem.

Recently, the discrete equilibrium optimization algorithm (DEOA) was proposed by Malik et al. [32] for solving the classification rule generation in discrete problems. The DEOA shows promising results compared to other swarm-based optimization algorithms [32]. Given the promising results of the DEOA, this paper adapts it for binary optimization problems. Thus, we propose a new binary version of DEOA called binary equilibrium optimization algorithm (BEOA) to optimize the CRs generation for ASA. The novelty of our work is related to the ASA model through a rule-based approach using BEOA [33].

The main contributions of this paper can be summarized as follows:

- A binary version of the discrete equilibrium optimization algorithm (DEOA) has been proposed and named BEOA. To adapt BEOA for binary optimization problems, we have used binary operators to update the position equation.
- The paper proposes a CRs generation approach using BEOA. Our approach generates an interpretable model with competitive classification accuracy compared to state-of-the-art approaches.
- The generated model allows the view and the study of the dataset terms' effect on the CRs. Thus, it may help improve the performance by using different NLP tools such as stemmers, stop word lists, and n-gram models.

The remainder of this paper is organized as follows: Sect. 2 presents the background of this work, including ASA and rule-based ASA. Section 3 exposes and discusses related works about ASA and rule-based classifier generation. Section 4 details the proposed approach. The experimental design, results, and discussion are reported in Sect. 5. Section 6 concludes this paper with perspectives on future works.

2 Background

2.1 Arabic Sentiment Analysis

The problem of sentiment analysis can be handled at three levels: document level, aspect level, and sentence level [34]. At the document level, the whole document is classified as to whether it expressed a positive or negative sentiment. The problem resolution at the aspect level is not limited to sentiment classification; it tries to identify the opinion target of the expressed sentiment [35]. At this level, aspect extraction is a preliminary phase. At the sentence level, we start by classifying sentences into subjective and objective ones, and then, the subjective sentences will be classified as positive or negative [34]. Two main approaches were adopted to resolve sentiment analysis in opinionated documents: corpus-based and lexicon-based. A set of hybrid methods is proposed in the literature to benefit from both corpus and lexicon approaches [36].

Work in sentiment analysis has achieved much advancement for English and other Indo-European languages; however, the situation for Arabic is different. To show this gap, the authors represent in Fig. 1 the number of researches, from Google Scholar, on sentiment analysis in English, Chinese, Spanish, Arabic, and French from 2000 to 2021. This lack of works dealing with ASA is added to Arabic's inherent characteristics related to the language specificities such as morphology, ambiguity, absence of capitalization, and Arabic dialects [14]. The ambiguity of Arabic comes, especially, from diacritic marks. These marks are absent in almost all Arabic documents, and Arabic native speakers can read such documents without these marks [37]. Arabic dialects are spoken forms of language used in daily conversation in the Arab world. These dialects are mainly used in social media and discussion forums [38]. The problem with dialects is the absence of standardized rules for writing them. Thus, it is challenging to have efficient NLP tools to handle all Arabic dialects [14].

2.2 Rule-Based Arabic Sentiment Analysis

The ASA can be considered a classification problem in data mining and NLP [34]. Rule-based classification (RBC), also

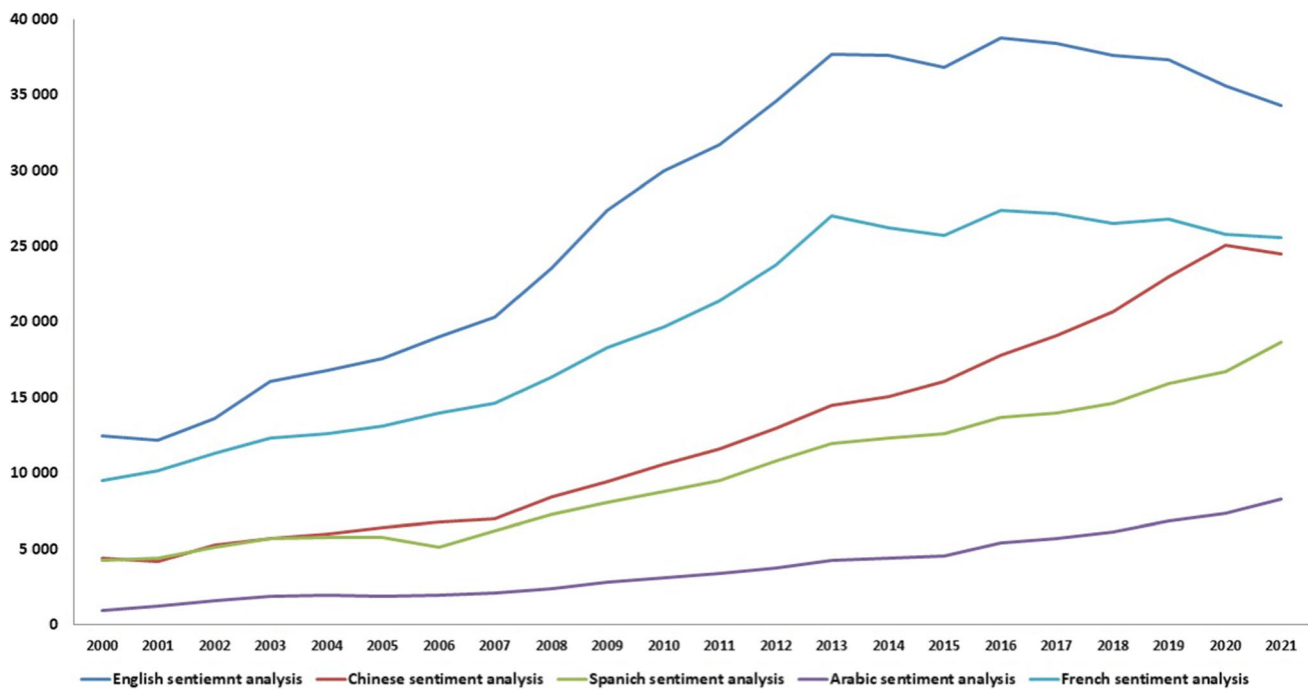


Fig. 1 Number of sentiment analysis researches on Arabic and other languages

called associative classification, is one of the data mining tasks introduced by Liu et al. [39] to extract CRs from datasets for decision support purposes in several domains. A rule-based classifier C is composed of CRs set ordered by their importance, $C = \{CR_1, CR_2, \dots, CR_N\}$, where N is the classifier size in terms of CRs number and CR_i is the i th CR in the classifier C .

We defined a CR for ASA as the association between a subset of words X and the class feature CF . When $X = \{w_1, w_2, \dots, w_p\}$ is called the rule’s antecedent, where w_i is the i th word in the word vector, the class feature CF is the consequent of the rule.

Let a word vector dataset $WVDS = \{wv_1, wv_2, \dots, wv_p\}$ where wv_i is the i th word vector in $WVDS$. Each word vector wv is represented by binary values corresponding to m words, $wv = \{a_1, a_2, \dots, a_m, CF\}$ where $a_i \in \{0, 1\}$; “1” indicates the presence of the word w_i in the document, and “0” indicates its absence. $CF \in \{0, 1\}$, where “0” means that the corresponding document is classified negative and “1” means that it is classified positive.

The rule-based classifier is used to classify a new unclassified document (word vector). For any new word vector wv_i , the CRs of the classifier are checked in order one by one. Once the classifier finds a classification rule CR_j that matches wv_i (i.e., the wv_i corresponds to the antecedent of the rule CR_j), it classifies wv_i according to the class label of the rule CR_j . If there is no rule matches wv_i , the classifier cannot classify this document and will assign it to the default class.

Let a classification model $C = \{R1, R2, R3, R4\}$ in Fig. 2a and a test set $T = \{wv_1, wv_2, wv_3, wv_4\}$ in Fig. 2b with the vocabulary $V = \{w_1, w_2, \dots, w_6\}$. In Fig. 2b, the classification result and the classification interpretation of each instance in T using the classifier C is presented.

3 Related Works

3.1 Arabic Sentiment Analysis

In the literature, several machine learning-based approaches have been proposed for ASA, and most of these approaches generate black-box models that are not understandable by the user. In contrast, few works propose interpretable classification models for ASA. The following literature review has grouped works into black-box and white-box models.

3.1.1 Black-Box Models

The work of Gamal et al. [40] used five machine learning algorithms for sentiment classification on the Arabic tweets dataset. The dataset comprises 151,000 tweets collected by a Twitter API using a set of Arabic keywords in both Modern Standard Arabic and dialectal Arabic. Experiments with ridge regression achieve the best results by 99.99% accuracy, outperforming support vector machines (SVM) and naïve Bayes (NB) classifiers.

Fig. 2 Instance classification by a rule-based classifier application

	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	Class
R ₁	1	0	1	1	0	1	0
	1	1	0	1	1	0	
R ₂	0	0	1	1	1	0	1
	1	0	1	0	0	0	
R ₃	0	1	0	1	1	0	1
	1	0	1	0	1	1	
R ₄	0	1	1	0	1	0	0
	1	0	1	0	1	1	

(a) Rule-based classifier

	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	Classification results	Classification interpretation
wv ₁	1	0	0	1	0	0	'0'	Classified by R ₁
wv ₂	1	0	1	0	0	1	'1'	Unclassified by R ₁ and Classified by R ₂
wv ₃	0	0	1	0	1	1	'1'	Unclassified by R ₁ and R ₂ , then Classified by R ₃
wv ₄	0	0	1	1	1	1	'0'	Unclassified by R ₁ , R ₂ and R ₃ , then Classified by R ₄ .

(b) Instances classification

The authors in [41] used two datasets: one publicly available OCA [33] and the second is ACOM collected from the Aljazeera website. In their experimental study, the authors implemented three well-known machine learning algorithms: SVM, NB, and K-NN. SVM and NB classifiers give the best results, while the K-NN performance depends on the corpus.

In SANA (sentiment analysis on newspapers comments in Algeria) [21], the authors created their corpus from Algerian Newspaper websites intended to study the opinion mining problem. For the experiments, SVM, NB, and K-NN machine learning methods were applied to SANA and OCA corpora to compare the results. The accuracy depends on the corpus, and NB with Bi-gram achieves the best performance.

The work [7] has studied people's sentiment toward the COVID-19 epidemic in India and worldwide. The authors use a bidirectional encoding representation for a transformer model on two datasets: The first is from Indian users and the second from users around the world. In the experimental study, the proposed approach achieves 94% accuracy.

A deep learning-based framework that allows the improvement of the accuracy of ASA is developed in [4]. The work has implemented deep learning models to represent Arabic text from the Twitter social network. The collected tweets are written in both Moroccan dialect and Modern Standard Arabic. In the experiments, the authors use three machine learning methods, i.e., NB, SVM, and

maximum entropy. Also, they investigate a deep learning approach using a convolutional neural network and long short-term memory models. The deep learning models outperform basic machine learning methods in the basic representation without text preprocessing. In addition, the authors observe the tremendous impact of preprocessing operations, such as the light stemmer, on increasing the performance results of all classifiers.

Machine learning-based sentiment polarity detection in seven languages, including Arabic, is studied in [19]. A set of n-gram models is tested for byte, character, and word level. Byte and character n-gram models outperform word n-gram models in almost all cases in the different languages. In contrast to other languages, the K-NN classifier on the OCA corpus performs the best results in Arabic.

Sentiment analysis of people against the COVID-19 pandemic from Twitter social media network is achieved in [42]. A dataset of 1500 tweets, 750 positives and 750 negatives, is collected and annotated by two experts. Three well-known machine learning methods were used to classify tweets into positive and negative classes. SVM classifier gives the best results. The obtained results prove the impact of preprocessing steps in increasing the accuracy.

From the presented black-box approaches, it is clear that very accurate models are developed. The developed models give very interesting classification results. Despite these important obtained results via black-box methods, the users

cannot understand how a model has assigned a document to a class or another.

3.1.2 White-Box Models (Rule-Based Sentiment Analysis)

White-box-based classification approach has been introduced to overcome the drawback of black-box classification methods in terms of model interpretability. The importance of the developed white-box models is to allow the users to explain their decisions. Two main techniques are used for generating white-box classification models, i.e., decision trees (DTs) and RBC.

DT-based classification is a white-box classification method commonly used in several studies. DTs provide a hierarchical decomposition of the training data starting with the root until the leaf nodes. Each node in the tree is labeled by word occurrences (features) in the document. The branches are labeled by the weight of the word in the document. Finally, the leaves are labeled by the class value. Harrag et al. [43] used a decision tree (ID3) for classifying Arabic text documents. The authors construct firstly a vector containing all words present in all documents of the training dataset. In the second stage, they select a subset of words from the constructed vector according to some criteria. Finally, weight is affected to each word in the vector. The experimental results using two different corpora show that the corpus documents' nature and specificity impact the classification performance.

In [44], the authors created a Sentiword lexicon from the used corpus and compared the DT with SVM and NB for a sentiment analyzer of Arabic YouTube pages using a gathered Corpus. The experimental results show the superiority of the NB algorithm. In [45], the authors aim to study a comparison of the DT, SVM, and NB for ASA on Twitter. Their work deals with Modern Standard Arabic. The experimental results show that DT outperforms the other techniques by obtaining 78% of the F-measure. However, in DTs, the training's simple change may significantly change the generated model [46].

Few RBC approaches have been used in ASA literature. The rule-based classifiers are based on CRs generated from the training dataset. A CR has the simple form IF-THEN to classify a document, making it easily understandable by the user. Thus, it enhances the interpretability of generated models.

The authors in [47] proposed a novel approach based on the rough set theory for uncertain, incomplete, and vague information classification and analysis. In this work, the methodology starts with text preprocessing, including tokenization, removing stop words and stemming, and the term weighting using TF-IDF. The authors have used a dataset of Egyptian tweets consisting of 4812 documents. In the experiments, the authors compare two types of reduct, full

reduct and object reduct, with two classifiers: majority voting and NB. The highest percentage of accuracy, 54%, is achieved by object reduct when applied over genetic reducer.

A framework for sentiment analysis from Twitter is developed on the base of a rough set theory paradigm for generating corpus-based rules [1]. The authors have used the rough set theory-based algorithms for rules generation, i.e., exhaustive, genetic, covering, and LEM2 (Learning from Examples Module 2). The framework implemented a novel rule induction algorithm to provide maximum coverage in Tweets classification. The performance of the proposed method, LEM2-CBR, is auspicious compared with similar approaches, despite the limited number of rules generated by the LEM2 model.

3.2 Rule-Based Classifier Generation

Currently, several methods are available for CRs generation from datasets. These methods use CRs set for classifying instances. We categorized these methods by the nature of rule generation and grouped them into two distinct groups. First, indirect methods generate the CRs using an intermediate model such as decision trees and association rule (AR) set. The latter group concerns direct methods that generate CRs directly from the dataset without using intermediate models.

DTs and random forests are commonly used as models for generating CRs because of their easy conversion into a set of CRs. C4.5, CART, and ID3 [48, 49] are the most well-known DT methods. However, in DTs, the training's simple change may produce a more significant change in the generated model [50].

Wang et al. [51] proposed a method called improved random forest-based rule extraction for rules extraction from a random forest tree. This method aims to derive CRs from a set of DTs generated by the random forest algorithm using a multi-objective evolutionary algorithm for optimizing the extracted CR set.

The CR generation from ARs operates in two phases: The first phase aims to generate AR set from a dataset using known algorithms such as Apriori [52]. An AR represents the relationships among features of the dataset; however, a CR represents the relationship between the dataset's features and the class feature. Then, the classification model is composed only of a set of CRs [39]. The second phase extracted a subset of CRs from the AR set, satisfying prespecified support and confidence values.

Liu et al. [39] proposed an algorithm that extracts a CR set from the AR set generated by the Apriori algorithm. Variants of the Apriori algorithm are used in [53] and [54]. The classification based on predictive association rules [55] generates the ARs by an exhaustive search and then ranks the generated rules to form the classifier.



However, this algorithm is expensive in term of run time. So, to reduce the run time, the authors in [56] proposed a classification based on a multiple association rules algorithm, in which the ARs are generated using CR-Tree and FP-Tree algorithms. Thabtah et al. [57] proposed an algorithm called multi-class classification based on association rules, in which the Tid-list method is used during the rule generation stage. Hadi et al. [58] developed the enhancement class association rules and fast associative classification algorithms by employing the exact match prediction method to predict unseen text in a Saudi press dataset. The experiments indicate that the enhancement class association rules classifier outperformed traditional classifiers (K-NN, SVM, DT, and NB) concerning to error rate, recall, and precision.

In these indirect CR generation approaches, the number of rules depends on the number of features in the dataset, which increases the run time of the algorithms, especially in the cases of big datasets. In addition, it generates classifiers with a large number of rules which complicate their interpretability. Then, the direct CR set generation approach can build classifiers whose size is independent of the dataset's features.

Hasanpour et al. [49] used an evolutionary population-based algorithm called harmony search to extract ideal CRs from the dataset. Ant-Miner, a swarm-based algorithm, is used to explore the entire search space and generates a set of CRs directly from the dataset [59]. To improve the quality of the Ant-Miner algorithm, Holden and Freitas [60] hybridized the ant colony optimization [61] algorithm with the particle swarm optimization algorithm [62]. Otero et al. [63] propose a new sequential covering strategy for Ant-Miner to mitigate the problem of rule interaction. The authors proposed in [64] a new algorithm called Ant-Miner_{PAE} to overcome the premature convergence to local optimum in the ant colony optimization algorithm.

DEOA has been recently proposed to tackle discrete optimization problems and demonstrate promising results. This paper proposes a binary version of DEOA to resolve the CR generation in ASA as a binary problem. The proposed binary version of DEOA is an optimization algorithm called BEOA. The proposed approach uses the OCA corpus to create a word vector for generating CR set as an interpretable classification model. Our novel approach optimizes the rule generation process from word vector by generating a small set of rules improving the classification model interpretability.

4 Proposed Approach

The proposed approach, as presented in Fig. 3, is carried out in four phases:

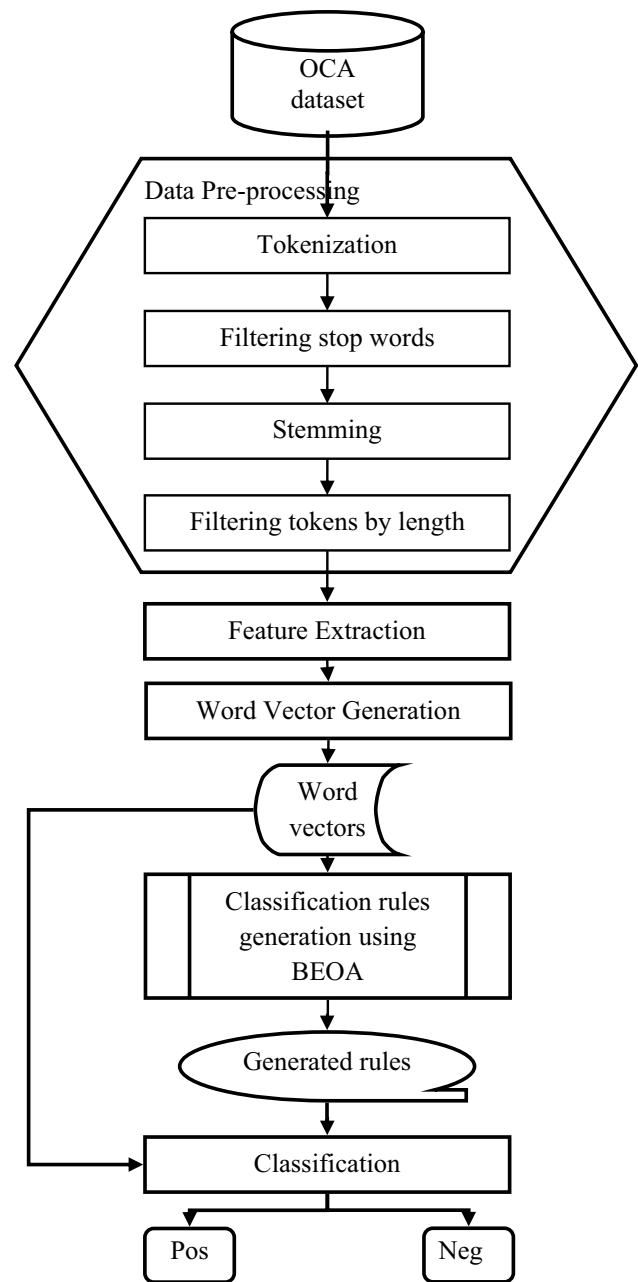


Fig. 3 Proposed rule-based Arabic sentiment analysis approach

- (i) Data preprocessing, including tokenization, filtering stop words, stemming, and filtering tokens by length;
- (ii) Feature extraction;
- (iii) Rule-based classifier generation using BEOA;
- (iv) The use of the rule-based classifier to sentiment classification as positive or negative.

4.1 Used Dataset

This work has used the OCA corpus [33] for experiments. The OCA dataset consists of 500 reviews for Arabic movies.

From the reviews, 250 have positive sentimental orientation, and 250 have negative sentimental orientation. For OCA annotation, the authors adopt the rating system of related websites to annotate the reviews. The annotation system considered reviews with more than three points as positive, while those with less than three points as negative. The authors consider reviews with three points as neutral and eliminate this category from their corpus.

The use of the OCA for this study is motivated by its prominent use in the community of ASA and its convenience for the current study. The first motivation is related to the prominent use of the OCA corpus since 2011 in the literature [65–69]. The second motivation is related to the homogeneity, the size of the documents, and the nature of their authors, i.e., the bloggers [41].

4.2 Data Preprocessing

The collected data from the web are noisy by nature. Preprocessing is a preliminary step in almost all text mining tasks. Preprocessing technique is used to reduce the size of documents and enhance the classification performance. This work uses the Rapidminer² toolkit and python language in this stage.

4.2.1 Tokenization

In tokenization, the words of text, also known as tokens, are obtained by splitting the text using white spaces and punctuation marks.

4.2.2 Stop Words Removal

Stop words are terms having no importance in document polarity identification. Eliminating these words helps in improving system performance by reducing the vector size. For the sake of simplicity, we adopt the list of Arabic stop words from the used toolkit Rapidminer [33, 70].

4.2.3 Stemming

Stemming is a common task in all NLP projects. We can differentiate stemming from root extraction. Stemming groups words in relation to semantics, while root extraction groups them according to a global meaning [71]. We used the full stemmer in this work to optimize our system performances. Thus, the word vector size will be reduced considerably. This work has employed the Arabic stemmer offered by the Rapidminer toolkit following some works using the OCA corpus [33, 70].

² <https://rapidminer.com/>

4.2.4 Filtering Tokens by Length

This step eliminates short terms, with less than two letters becoming symbols. Long terms, having more than 25 letters, have no vital polarity information are also eliminated in this phase.

4.3 Feature Extraction

Feature extraction aims to find the most suitable features for text classification. Extracted features may incorporate relevant knowledge for sentiment analysis, including semantic, commonsense, syntactic, and sentiment words [72]. This work used the uni-gram model for feature extraction, following the work [73] when the uni-gram model performs the best among other evaluated models.

4.4 Word Vector Generation

The text is mapped in this step to a vector representation. This representation facilitates the machine handling of textual data [19]. Several vector representation models are proposed in the literature [74, 75]. We used the binary term occurrence (BTO) in this work, which we estimate the most suitable for our work for its promising results in [70]. In the BTO model, the vector space corresponds to the corpus's total number of unique words. Thus, a word is represented by “1” if it is present in the document and by “0” otherwise.

$d_i = (w_{i1}, w_{i2}, \dots, w_{ik}) / w_{ij} \in \{0, 1\}$, where k represents the vocabulary, i.e., the set of all words in the used dataset. w_{ij} refers to the presence “1” or the absence “0” of the word “ j ” in the document “ i ,” in other terms whether this word contributes to the representation of this document [19]. This phase produces a vector of 698 attributes.

4.5 BEOA-Based Rule Generation for Arabic Sentiment Analysis

In this section, we present the proposed classification rule-based ASA approach. For this, the BEOA is proposed and used to generate the CRs. After preparing the word vector from the original dataset, i.e., OCA, the word vector is analyzed for generating a rule-based classifier model to classify new documents. The proposed approach works as follows: A new dataset is generated from the word vector at each iteration, containing only the class instances with the highest number of instances. BEOA is called iteratively to generate one CR from the new dataset in every iteration. After that, all instances covered by the generated rule in the new dataset are removed. This process must be repeated until the word vector dataset becomes empty. Algorithm 1 describes the main steps of classification rules generation using BEOA. Algorithm 2 gives the BEOA details.



Table 1 Particle's encoding

Words	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
Word selection in CR (CA)	1	0	1	0	1	0	1	1	1	0
Word value in CR (PA)	0	1	0	0	0	1	1	0	1	1

Due to its excellent performance in solving rule generation problems, the authors in [32] have studied DEOA to adapt it for ASA problems. Algorithm 1 presents the pseudo-code of the proposed BEOA used to generate a CR from the word vector dataset. The working principle of BEOA is described as follows:

4.6 Particle Position Encoding and Rule Representation

There are two different approaches for encoding the particle's position: Michigan and Pittsburgh. In the Michigan approach, a particle's position encodes only one rule, while in the Pittsburgh approach, the particle's position encodes a set of rules. However, the Pittsburgh approach has a complicated encoding that needs more computational time memory [76]. So, in this paper, the Michigan method [77] is used, where each particle searches a single CR from the training dataset for a selected class. The CR has

$$\text{Fitness} = \begin{cases} 1 & \text{if } \text{All_wv} = \text{wv_Cor_Clas} \\ \frac{\text{All_wv}}{(\text{All_wv} - \text{wv_Cor_Clas}) * (\text{All_wv} + \text{wv_Cov})} & \text{else } \text{All_wv} \neq \text{wv_Cor_Clas} \end{cases} \quad (1)$$

the form *IF-THEN*; the *IF* part, called the antecedent, is a conjunction of conditions in the form (word = 0 or 1). The value "1" indicates the presence of the word in the vector representing the document, while the value "0" means its absence. So, only the values "0" or "1" in the vector represent the particle's position, making the rule generation a binary optimization problem. The *THEN* part is a consequent that corresponds to the prediction class. In our approach, the class is previously selected (as presented in Algorithm 1). Hence, the particle should include only the *IF* part of the rule.

Two *d*-dimensional binary vectors are used for a particle's structure, where *d* is the number of words in the training word vector dataset. In the first vector, if the *i*th value is "1," then the *i*th word is selected in the rule, and if the value is "0," the *i*th word is not selected in the rule. The second vector represents the value of each selected word in the rule. If the *i*th value is "1," then the *i*th word is present in the word vector (document), and if the value "0," the word is absent in the word vector. Table 1 presents a detailed example.

The position of Table 1 represents the following CR $\{w_1=0$ and $w_3=0$ and $w_5=0$ and $w_7=1$ and $w_8=0$ and $w_9=1\} \Rightarrow Y$. This rule means that if the words $w_1, w_3, w_5,$ and w_8 are absent in the document and the words w_7 and w_9 are present in the document, then the class of the document is *Y*, and thus, *Y* is the class of the considered rule.

4.7 Particle Initialization

Random uniform initialization is used in this work, in which the initial positions of all particles are randomly scattered in the search space. The two vectors of the particle are randomly initialized by 0 or 1.

4.8 Fitness Function for Position Evaluation

Because the particle's position represents a CR, the quality of each CR is related to the number of correctly classified word vectors and the number of covered word vectors in the training dataset by the rule. Then, the used fitness function is as follows in Eq. 1.

where All_wv is the number of all word vectors in the dataset, wv_Cor_Clas is the number of word vectors correctly classified by the CR, and wv_Cov is the number of word vectors covered by the rule.

4.9 Position Updating

DEOA is a discrete metaheuristic that contains discrete valued operators (addition, multiplication, subtraction, and division) in the position update equation, while the proposed BEOA is a binary metaheuristic. Then, we must modify the operators in the update position equation of BEOA to adapt it to binary optimization problems. For this, new binary operators are used. Equation (2) presents the position update equation of each particle position in BEOA.

$$X_{i+1} = \begin{cases} X_{eq} + \frac{G}{\lambda} & \text{if } F \geq 0 \\ X_{eq} + (X_i - X_{eq}) & \text{else } F < 0 \end{cases} \quad (2)$$

where X_i and X_{i+1} are the current and the new position vectors of the particle, respectively, and X_{eq} is the equilibrium position randomly chosen from the equilibrium pool ($P_{eq, pool}$)

vector calculated using Eqs. (3) and (4). F is the exponential parameter vector used to balance exploration and exploitation, F is a vector of real values calculated using Eq. (6). G and λ are binary vectors and have the same structure as X_i . G is the generation rate parameter vector calculated using Eqs. (7) and (8), and λ is a random vector.

$$X_{eq_pool} = \{X_{eq0}, X_{eq1}, X_{eq2}, X_{eq3}, X_{ave}\} \tag{3}$$

$$X_{ave} = \text{Average_Equilibrium_Vector}(Ceq1, Ceq2, Ceq3, Ceq4) \tag{4}$$

where `Average_Equilibrium_Vector()` is the procedure presented in Algorithm 3.

$$t = \left(1 - \frac{\text{iter}}{\text{Max_iter}}\right)^{\left(a_2 \frac{\text{iter}}{\text{Max_iter}}\right)} \tag{5}$$

where $iter$ and Max_iter are the current and maximum iterations time, respectively, a_2 is another tuned parameter of the algorithm.

$$F = a_1 * \text{sign}(r - 0.5)(e^{-\lambda t} - 1) \tag{6}$$

where a_1 is a tuned parameter, r is a random vector in the interval $[0,1]$, t is the time, and it is decreased with the number of iterations, such as presented in Eq. (5). The operators “-” and “*” in Eq. (6) are real-valued.

$$G = \begin{cases} G_0 & \text{if } F > 0 \\ |1 - G_0| & \text{if } F \leq 0 \end{cases} \tag{7}$$

$$G_0 = X_{eq} - (\lambda * X_i) \tag{8}$$

where F is defined in Eq. (6) and $|*|$ is the absolute value function.

The operators “+,” “-,” “*,” and “/” in Eqs. (2) (4), and (7) are binary operators that operate between vectors. These operators are defined as follows:

Let two vectors X and Y that are the same structure of the vector X_i .

$$X - Y = \begin{cases} Y & \text{if } X \neq Y \\ 0 & \text{if } X = Y \end{cases} \tag{9}$$

$$X + Y = \begin{cases} X & \text{if } X \neq 0 \text{ and } Y = 0 \\ Y & \text{if } X = 0 \text{ and } Y \neq 0 \\ X \text{ OR } Y \text{ randomly} & \text{if } X \neq 0 \text{ and } Y \neq 0 \\ 0 & \text{if } X = 0 \text{ and } Y = 0 \end{cases} \tag{10}$$

$$X/Y = \begin{cases} X & \text{if } Y = 0 \\ 0 & \text{if } X = 0 \text{ and } Y = 1 \\ X \text{ OR } Y \text{ randomly} & \text{if } X = 1 \text{ and } Y = 1 \end{cases} \tag{11}$$

$$X * Y = \begin{cases} 0 & \text{if } X = 0 \text{ OR } Y = 0 \\ X \text{ OR } Y \text{ randomly} & \text{else} \end{cases} \tag{12}$$

Algorithm 1. Classification rules generation using BEOA
Input: Word Vector dataset $WV[N, M]$ with N vectors and M words in each vector
Output: Set of classification rules RS (rule-based classifier)
Begin
 $RS \leftarrow \emptyset$ //Initialize the rules set RS as empty.
while (WV dataset is not empty)
 $C \leftarrow$ the class name with the highest number of instances in the dataset WV
 $NWV \leftarrow \emptyset$ //Create a new empty word vector dataset (NWV)
 // Insert all vectors with the class C from WV to NWV
 for($i=1$ to size of WV)
 if($WV[i,M] = C$)
 $NWV \leftarrow NWV + WV[i]$ //insert $WV[i]$ to NWV
 $WV \leftarrow WV - WV[i]$ //Delete $WV[i]$ from WV
 end if
 end for
 while(NWV in not empty)
 $CR \leftarrow \text{BEOA}(NWV,C)$ // the algorithm $\text{BEOA}(NWV,C)$ is //presented in Algorithm 2.
 $RS \leftarrow RS + CR$ // Add CR in RS
 //Delete all vectors from NWV that covered by CR
 For($j=1$ to the size of NWV)
 if($NWV[j]$ match the rule CR)
 $NWV \leftarrow NWV - NWV[j]$ //Delete $NWV[j]$ from NWV
 end if
 end for
 end while
end while
Return RS
End

Algorithm 1: Classification rules generation using BEOA

Algorithm 2. BEOA (NWV,C) // Binary Equilibrium Optimization Algorithm**Input:** population size (pop-size), maximum number of iterations (tmax), BEOA parameters: a1, a2 and GP**Output:** classification rule**Begin**

```

Initialize the particle's positions randomly  $p_i(i=1,2,3, \dots, \text{pop-size})$ 
Evaluate the fitness of all particles positions using Equation (1)
 $Ceq_1 \leftarrow p_1; Ceq_2 \leftarrow p_2; Ceq_3 \leftarrow p_3; Ceq_4 \leftarrow p_4;$ 
for (t=1 to tmax)
    // Update the equilibrium vector  $C_{eq}$ 
    for(i=1 to pop-size)
        if (fitness( $p_i$ )>fitness( $Ceq_1$ ))
             $Ceq_1 \leftarrow p_i$ 
            fitness( $Ceq_1$ )  $\leftarrow$  fitness( $p_i$ )
        elseif (fitness( $p_i$ )>fitness( $Ceq_2$ ))
             $Ceq_2 \leftarrow p_i$ 
            fitness( $Ceq_2$ ) = fitness( $p_i$ )
        else if (fitness( $p_i$ )>fitness( $Ceq_3$ ))
             $Ceq_3 \leftarrow p_i$ 
            fitness( $Ceq_3$ ) = fitness( $p_i$ )
        else if (fitness( $p_i$ )>fitness( $Ceq_4$ ))
             $Ceq_4 \leftarrow p_i$ 
            fitness( $Ceq_4$ )  $\leftarrow$  fitness( $p_i$ )
        end if
    end if
end if
end for
 $Ceq_{avg} \leftarrow$  Average_Equilibrium_vector( $Ceq_1, Ceq_2, Ceq_3, Ceq_4$ )// The procedure
//Average_Equilibrium_Vector() is presented in Algorithm 3.
 $Ceq = [Ceq_1, Ceq_2, Ceq_3, Ceq_4, Ceq_{avg}]$ 
Accomplish the memory saving
//After the second iteration, the fitness value of the current particle in the current iteration is
//compared to that of the previous iteration and will be overwritten if it achieves a better fitness
Calculate t using the Equation (6)
for(i=1 to pop_size)
    Choose randomly one candidate vector from Ceq
    Generate randomly two binary vectors r and  $\lambda$  similar to  $p_i$ 
    Calculate F using the equation (5)
    Calculate G using the equation (7)
    Update the particle  $p_i$  using the equation (2)
Check the feasibility of the new solution  $p_i$ 
Evaluate the fitness of the new position using Equation (1)
end for
end for

```

Algorithm 2 : BinaryEquilibriumOptimization Algorithm


```

Algorithm 3. Average_Equilibrium_Vector (Ceq1, Ceq2, Ceq3, Ceq4)
Output: Ceqavg
Begin
  s = Ceq1.CA + Ceq2.CA + Ceq3.CA + Ceq4.CA
  if (s ≥ 2)
    Ceqavg.CA = 1
    if (Ceq1.CA ≠ 0)
      Ceqavg.PA = Ceq1.PA
    else
      if (Ceq2.CA ≠ 0) Ceqavg.PA = Ceq2.PA
      else
        Ceqavg.PA = Ceq3.PA
      end if
    end if
  else
    Ceqavg.CA = 0
  end if
End.

```

Algorithm 3: Average Equilibrium Vector Algorithm

5 Experimental Study

This section is devoted to the performance evaluation of the proposed approach. Firstly, we describe the experimental design of the proposed approach in terms of evaluation criteria and the extracted rules from the used dataset. Then, the results of our approach are compared with the literature approaches in terms of the evaluation criteria. Finally, the obtained results are discussed and interpreted.

5.1 Experimental Design

This section aims to study the results of our approach and compare it with other well-known approaches on the OCA corpus. The corpus has been split into two partitions: 80% is used for training, while 20% is used for testing. The proposed approach was developed and implemented using Weka and Rapidminer toolkits in addition to Python and Java programming languages.

5.1.1 Classification Rules Extraction

The BEOA extracts a set of rules from the used dataset for document classification. For each extracted rule, the rule antecedent, the class, the value of rule coverage, the rule accuracy, and the number of terms were recorded. The rule coverage counted as the number of instances in the training dataset satisfying the antecedent, but not the rule's consequent. The rule accuracy counted as the percentage of the classified instances (instances satisfying the antecedent and the consequent of the rule simultaneously) to the rule's coverage.

5.1.2 Evaluation Criteria

The proposed approach is evaluated based on four metrics: classifier accuracy, coverage, and complexity in terms of the number of rules and the average length of rules of the classifier.

- **Classifier accuracy:** is the primarily used metric; it is defined as the ratio [78, 79] of the number of correctly classified instances divided by the total number of instances in the test dataset, such as presented in Eq. (13).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + E + \text{FN}} \quad (13)$$

- **Precision:** precision is calculated using Eq. (14).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (14)$$

- **Recall:** recall is calculated using Eq. (15).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (15)$$

- **Classifier complexity:** measured by the classifier size (number of rules in the classifier).

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

5.2 Extracted Rules From the OCA Dataset

The proposed approach generates thirteen CRs from the OCA dataset. Table 2 lists the generated rules. The last

column represents the number of terms (words) in the rule antecedent.

Table 2 shows that all rules have got considerably high accuracy value, and the average number of terms in the rule set is 14,15. The number of rules and terms in the rule are two metrics of the rule's interpretability.

A CR can be easily read; for example, the sixth rule can be read as follows; *IF* the document does not contain the words "أمل" and "جنني" and "رومانسي" and "سوك" and "مبقر" and "مرض" and "نزل" and "يوسف" and does contain the word "رأس," *THEN* it will be classified in the negative class.

On observing and analyzing the rules listed in Table 2, it can be seen that the class of the comment is significantly dependent on the words present in the rules, such as "ذهل", "فرح", "شقف", "شعب", "فجر", "ضغظ", "ضخم", and other.

Despite the important obtained performance compared to the literature, we observe, from the antecedents of CRs, that an important number of words used as features of the model are non-Arabic words, i.e., non-dictionary words. We report this to the nature of the used stemmer that, once it generates the word's root, it does not seek for obtained stems in the Arabic dictionary for validation. So the interpretability of our model allows us to fix the source of this limitation to deal with it later. The obtained model is constituted of thirteen CRs generated from a word vector of 698 features (words). This model is very acceptable in terms of interpretability.

5.3 Performance Comparison with Other Approaches

The goal of this study is to analyze and compare the performance of our approach. We have evaluated the performance of the proposed approach by comparing it with some popular classification algorithms, including white-box and black-box methods. Therefore, we carried out a set of white-box methods, i.e., PART [80], RIPPER [81], OneR [82], C4.5 [48], and REPTree [83] on WEKA toolkit [84]. The three black-box algorithms SVM [85], K-NN [86], and NB [87] are also implemented through the same platform and used for results comparison.

Table 3 shows the accuracy, recall, precision, and classifier size (number of rules obtained by the tested algorithms). Analyzing accuracy, in the black-box approaches, NB has obtained the best results. For white-box approaches, it is clear that the model generated by our approach achieves the best results, attaining 84% of accuracy, and vastly outperforms the state-of-the-art methods by more than 10% of accuracy.

The last column in Table 3 is dedicated to measuring the number of generated rules (for white-box algorithms). This number allows the reader to focus on the interpretability of

classification models. From this table, we can see that results obtained by RIPPER and PART are concurrent to those obtained by our approach in terms of recall and precision. However, our approach achieved a better trade-off between accuracy and interpretability by using the intelligent BEOA for CRs generation, a characteristic we do not see in other rule-based methods.

6 Results and Discussion

In several domains, like medical diagnosis, industrial diagnosis [88–90], and climate forecast [91], the studies concluded that the rule-based classification is the appropriate method to facilitate the interpretability of the classification decision [92].

This work has presented a novel rule-based classification approach for generating an ASA model from the OCA corpus. The CRs have been intelligently generated by using the BEOA metaheuristic. Therefore, the extracted set of understandable rules in Table 3 can easily guide decision makers or supervisors in selecting the crucial words that influence the writers' sentiment and comportment. We believe that the compact classification model in Table 3 is due to the intelligence and the optimization property of the BEOA metaheuristic in the global ASA proposed approach. So, the current study demonstrated that the rule-based classifier is still effective for sentiment analysis.

Despite that, the obtained accuracy by our model, i.e., 84%, is higher than all other white-box models as well as black-box models, and his precision is 100%, the accuracy stays lower than 85%, which requires further improvement in the rule generation process.

7 Conclusion and Perspectives

Sentiment analysis, also known as opinion mining, is a relatively recent field at the crossroad of data mining natural language processing and computational linguistics. The field had considerable achievements in recent years in English and other Indo-European languages, but works in low-resourced languages such as Arabic are still in their beginning. The current work comes in this context to contribute to reducing this gap.

This paper worked on the Arabic sentiment analysis problem by applying a new binary metaheuristic to generate a rule-based classification model on the OCA corpus. To obtain the best performance using the proposed approach, a set of preprocessing steps were carried out on the used corpus employing the Rapidminer toolkit and the python programming language.

Although black-box classification algorithms give high classification accuracy, they suffer from a critical limitation: Their generated classification models are not understandable by the users. This research has focused on using a metaheuristic algorithm in rule-based classifier generation. We proposed the new population-based algorithm BEOA for generating the best set of CRs that can be used as an accurate classifier. The proposed algorithm is implemented using the Weka toolkit and the Java programming language. Hence, the superior accuracy of our proposed approach compared to other literature approaches.

Using our proposed model, a set of thirteen (13) rules is generated from the OCA corpus for Arabic document polarity classification. On average, 14,15 terms were obtained per rule, which is considered acceptable knowing other rule-based classification studies. Therefore, the set of a few short rules could be beneficial for finding the causes of text writer sentiment from a rule usage point of view. In each classification rule, the words in the rule's antecedent are considered as causes of the consequent (class) of the rule.

We observed from the obtained model that an important number of terms/features are non-Arabic words, although these features enhance the model accuracy. In the word vector of our approach, we prospect the improvement of the preprocessing algorithms such as stemmers, stop words lists, and feature selection methods. In future work, a new fitness function that considers the rule's interpretability may be proposed to give more robustness to our approach. Another improvement may be obtained by using other optimization algorithms for the rule generation problem.

Concerning the proposed BEOA, on the one hand, we can improve its optimization performance by combining it with other metaheuristics to produce a new hybrid rule generation algorithm. This hybrid algorithm may involve new exploitation and exploration strategies for finding accurate rules. On the other hand, we can check the rule set by an additional step to improve their accuracy, for example, rule pruning by adding new features or removing features from rules, i.e., doing a local search around the rules. Finally, the proposed approach has proved competitive with state-of-the-art approaches. However, the large number of attributes can decrease the performance of the proposed approach in terms of classifier size and time consumption. Thus, a preprocessing step should be considered on corpora before applying the rule generation algorithm to remove irrelevant and redundant features.

References

1. Asghar, M.Z.; Khan, A.; Khan, F.; Kundi, F.M.: RIFT: a rule induction framework for twitter sentiment analysis. *Arab J Sci Eng* **43**(2), 857–877 (2018). <https://doi.org/10.1007/s13369-017-2770-1>
2. Liu, B.: *Sentiment Analysis Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, Cambridge (2015)
3. Liu, B.: Sentiment analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol.* **5**(1), 1–167 (2012). <https://doi.org/10.2200/S00416ED1V01Y201204HLT016>
4. Oussous, A.; Benjelloun, F.Z.; Lahcen, A.A.; Belfkih, S.: ASA: A framework for arabic sentiment analysis. *J. Inf. Sci.* **46**(4), 544–559 (2020). <https://doi.org/10.1177/0165551519849516>
5. Cherif, W.; Madani, A.; Kissi, M.: Towards an efficient opinion measurement in arabic comments. *Procedia Comput. Sci.* **7**, 122–129 (2015). <https://doi.org/10.1016/j.procs.2015.12.057>
6. Harb, A.; Gérard, D.; Michel, P.: Détection d'opinion: apprenons les bons adjectifs. In : *Atelier FODOP*, pp. 59–66 (2008)
7. Singh, M.; Kumar, A.; Shivam, J.: Sentiment analysis on the impact of coronavirus in social life using the BERT model. *Soc. Netw. Anal. Min.* (2021). <https://doi.org/10.1007/s13278-021-00737-z>
8. Guellil, I.; Adeel, A.; Azouaou, F.; Benali, F.; Hussain, A.: Arabizi sentiment analysis based on transliteration and automatic corpus annotation. In: *9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2018, pp. 335–341. <https://www.aclweb.org/anthology/W18-6249/>
9. Duwairi, R.M.; Alfaqeh, M.; Wardat, M.; Alrabadi, A.: Sentiment analysis for Arabizi text. In: *7th International Conference on Information and Communication Systems, ICICS*, 2016, pp. 127–132, doi: <https://doi.org/10.1109/IACS.2016.7476098>.
10. Guellil, I.; Azouaou, F.; Benali, F.; Hachani, A.E.; Mendoza, M.: The role of transliteration in the process of arabizi translation/sentiment analysis. In: *Abd Elaziz, M.; Al-qaness, M. A. A.; Ewees, A. A.; Dahou, A. (Eds) Recent advances in NLP: The Case of Arabic Language*. Springer International Publishing, Cham, 2020, pp. 101–128.
11. Masmoudi, A.; Khmekhem, M.E.; Khrouf, M.; Belguith, L.H.: Transliteration of Arabizi into arabic script for tunisian dialect. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* (2019). <https://doi.org/10.1145/3364319>
12. Al-badrashiny, M.; Eskander, R.; Habash, N.; Rambow, O.: Automatic transliteration of romanized dialectal Arabic. In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 2014, pp. 30–38.
13. Masmoudi, A.; Habash, N.; Khmekhem, M.E.; Estève, Y.; Belguith, L. H.: Arabic transliteration of romanized tunisian dialect text: a preliminary investigation. In: *Computational Linguistics and Intelligent Text Processing. CICLING 2015. Lecture Notes in Computer Science*, 2015, pp. 608–619, doi: <https://doi.org/10.1007/978-3-319-18111-0>.
14. Rahab, H.; Djoudi, M.; Zitouni, A.: Sentiment analysis of arabic documents: main challenges and recent advances. In: *Pinarbasi, F.; Taskiran, N. (Eds) Natural Language Processing for Global and Local Business*. IGI Global, pp. 307–331, (2021).
15. Mubarak, H.; Abdelali, A.; Darwish, K.; Eldesouki, M.; Samih, Y.; Sajjad, H.: A System for diacritizing four varieties of Arabic. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, September, pp. 217–222, doi: <https://doi.org/10.18653/v1/D19-3037>.
16. Fadel, A.; Tuffaha, I.; Al-jawarneh, B.; Al-Ayyoub, M.: Arabic text diacritization using deep neural networks. In: *International Conference on Computer Applications & Information Security*, pp. 2–9 (2019). <https://doi.org/10.1109/CAIS.2019.8769512>.
17. Zalmout, N.; Habash, N.: Joint Diacritization, Lemmatization, Normalization, and Fine-Grained Morphological Tagging, *arXiv Prepr arXiv191002267*, 2019.



18. Oueslati, O.; Cambria, E.; Ben HajHmida, M.; Ounelli, H.: A review of sentiment analysis research in Arabic language. *Fut. Gen. Comput. Syst.* **112**, 408–430 (2020). <https://doi.org/10.1016/j.future.2020.05.034>
19. Graovac, J.; Mladenović, M.; Tanasijević, I.: NgramSPD: exploring optimal N-gram model for sentiment polarity detection in different languages. *Intell. Data Anal.* **23**(2), 279–296 (2019)
20. Alnemer, L.; Alammouri, B.; Alsakran, J.; Ariss, O.E.: Enhanced classification of sentiment analysis of arabic reviews. *Springer* **2**, 210–220 (2019). <https://doi.org/10.1007/978-3-030-12839-5>
21. Rahab, H.; Zitouni, A.; Djoudi, M.: SANA: Sentiment analysis on newspapers comments in Algeria. *J. King Saud. Univ. Comput. Inf. Sci.* **33**(7), 899–907 (2019). <https://doi.org/10.1016/j.jksuci.2019.04.012>
22. Rahab, H.; Zitouni, A.; Djoudi, M.: ARAACOM: Arabic Algerian corpus for opinion mining, *arXiv*. 2020.
23. Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; Stede, M.: Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **37**(2), 267–307 (2011). https://doi.org/10.1162/COLI_a_00049
24. Abdul-Mageed, M.; Diab, M.: Toward building a large-scale arabic sentiment lexicon. In: 6th International Global Wordnet Conference, pp. 18–22 (2012).
25. El-Beltagy, S. R.: NileULex: a phrase and word level sentiment lexicon for egyptian and modern standard Arabic. In: Proceedings of Tenth International Conference on Language Resources and Evaluation (LREC), pp. 2900–2905 (2016).
26. Mataoui, M.; Zelmati, O.; Boumechache, M.: A proposed lexicon-based sentiment analysis approach for the vernacular algerian arabic. *Res. Comput. Sci.* **110**, 55–70 (2016)
27. Abdelhameed, H.J.; Hernández, S.M.: Sentiment analysis of arabic tweets in sudanese dialect. *Int. J. New Technol. Res.* **5**(6), 17–22 (2019)
28. Bahrami, M.; Amiri, M.J.; Mahmoudi, M.R.; Koochaki, S.: Modeling caffeine adsorption by multi-walled carbon nanotubes using multiple polynomial regression with interaction effects. *J. Water Health* **15**(4), 526–535 (2017)
29. Pan, J.-J.; Mahmoudi, M.R.; Baleanu, D.; Maleki, M.: On comparing and classifying several independent linear and non-linear regression models with symmetric errors. *Symmetry (Basel)* **11**(6), 820 (2019)
30. Maleki, M.; Mahmoudi, M.R.: Two-piece location-scale distributions based on scale mixtures of normal family. *Commun. Stat. Methods* **46**(24), 12356–12369 (2017)
31. Abbasi, A.R.; Mahmoudi, M.R.: Application of statistical control charts to discriminate transformer winding defects. *Electr. Power Syst. Res.* **191**, 106890 (2021)
32. Mahdi, M. M.; Hichem, H.: Efficient sequential covering strategy for classification rules mining using a discrete equilibrium optimization algorithm. *J. King Saud. Univ. Inf. Sci.* (2021)
33. Rushdi-Saleh, M.; Martín-Valdivia, M.T.; Ureña-López, L.A.; Perea-Ortega, J.M.: OCA: Opinion corpus for Arabic. *J. Am. Soc. Inf. Sci. Technol.* **62**(10), 2045–2054 (2011). <https://doi.org/10.1002/asi.21598>
34. Liu, B.: *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, 2nd edn. Cambridge University Press, Cambridge (2020)
35. Jayasekara, I.R.; Wijayanayake, W.M.J.I.: Opinion mining of customer reviews: feature and smiley based approach. *Int. J. Data Min. Knowl. Manag. Process* **6**(1), 1–11 (2016)
36. Mohammad, S.M.; Salameh, M.; Kiritchenko, S.: Sentiment lexicons for arabic social media. In: Tenth International Conference on Language Resources and Evaluation (LREC), pp. 33–37 (2016).
37. Darwish, K., et al.: A panoramic survey of natural language processing in the Arab world. *Commun. ACM* **64**(4), 72–81 (2021). <https://doi.org/10.1145/3447735>
38. Habash, N.Y.: Introduction to Arabic natural language processing. *Synth. Lect. Hum. Lang. Technol.* **3**(1), 1–187 (2010)
39. Liu, B.; Hsu, W.; Ma, Y.: Integrating classification and association rule mining. *Kdd* **98**, 80–86 (1998)
40. Gamal, D.; Alfonse, M.; Salem, A.M.: Twitter benchmark dataset for arabic sentiment analysis. *Int. J. Mod. Educ. Comput. Sci.* **11**(January), 33–38 (2019). <https://doi.org/10.5815/ijmecs.2019.01.04>
41. Mountassir, A.; Benbrahim, H.; Berraba, I.: Sentiment classification on Arabic corpora. A preliminary cross-study. *Int. Conf. Innov. Tech. Appl. Artif. Intell.* **16**(1), 259–272 (2013). <https://doi.org/10.3166/dn.16.1.73-96>
42. Ayadi, R.; Shahin, O.R.; Ghorbel, O.; Alanazi, R.; Saidi, A.: Sentiment analysis of COVID-19 tweets: impact of pre-processing step. *Int. J. Comput. Sci. Netw. Secur.* **21**(3), 206–211 (2021)
43. Harrag, F.; El-Qawasmeh, E.; Pichappan, P.: Improving arabic text categorization using decision trees. In: 2009 First International Conference on Networked Digital Technologies, pp. 110–115 (2009).
44. Elawady, R.M.; Barakat, S.; Nora, M.E.: Sentiment analyzer for arabic comments. *Int. J. Inf. Sci. Intell. Syst.* **3**(4), 73–86 (2014)
45. Altawaier, M.M.; Tiun, S.: Comparison of machine learning approaches on arabic twitter sentiment analysis. *Int. J. Adv. Sci. Eng. Inf. Technol.* **6**(6), 1067–1073 (2016)
46. Han, J.; Pei, J.; Kamber, M.: *Data Mining: Concepts and Techniques*. Elsevier, Third (2011)
47. Al-Radaideh, Q.A.; Twaik, L.M.: Rough set theory for Arabic sentiment classification. *Int. Conf. Fut. Int. Things Cloud* **2014**, 559–564 (2014)
48. Salzberg, S.L.: *C.4: Programs for machine learning by j. ross quinlan*. morgan kaufmann publishers, inc, 1993. *Mach Learn.* **16**(3), 235–240 (1994)
49. Hasanpour, H.; Meibodi, R.G.; Navi, K.: Improving rule-based classification using Harmony Search. *PeerJ Comput. Sci.* **5**, e188 (2019). <https://doi.org/10.7287/peerj.preprints.27634v1>
50. Pires, M.; Silva, N.; Rocha, L.; Meira, W.; Ferreira, R.: Efficient parallel associative classification based on rules memorization. In: International Conference on Computational Science, pp. 31–44 (2019).
51. Wang, S.; Wang, Y.; Wang, D.; Yin, Y.; Wang, Y.; Jin, Y.: An improved random forest-based rule extraction method for breast cancer diagnosis. *Appl. Soft. Comput.* **86**, 105941 (2020)
52. Hegland, M.: The apriori algorithm—a tutorial. *Math Comput Imaging Sci Inf Process*, pp. 209–262 (2007).
53. Jovanoski, V.; Lavrač, N.: Classification rule learning with APRI-ORI-C. In: Portuguese Conference on Artificial Intelligence, pp. 44–51 (2001).
54. Scheffer, T.: Finding association rules that trade support optimally against confidence. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 424–435 (2001).
55. Yin, X.; Han, J.: CPAR: Classification based on predictive association rules. In: Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 331–335 (2003).
56. Li, W.; Han, J.; Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: Proceedings 2001 IEEE International Conference on Data Mining, pp. 369–376 (2001).
57. Thabtah, F.; Cowling, P.; Peng, Y.: MCAR: multi-class classification based on association rule. In: The 3rd ACS/IEEE International Conference on Computer Systems and Applications, p. 33 (2005).
58. Hadi, W.; Aburub, F.; Alhawari, S.: A new fast associative classification algorithm for detecting phishing websites. *Appl. Soft. Comput.* **48**, 729–734 (2016)
59. Parpinelli, R.S.; Lopes, H.S.; Freitas, A.A.: Data mining with an ant colony optimization algorithm. *IEEE Trans. Evol. Comput.* **6**(4), 321–332 (2002)
60. N. Holden and A. A. Freitas, A hybrid PSO/ACO algorithm for discovering classification rules in data mining, *J. Artif Evol Appl*, vol. 2008, 2008.



61. Dorigo, M.; Birattari, M.; Stutzle, T.: Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**(4), 28–39 (2006)
62. Kennedy, J.; Eberhart, R.: Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*, vol 4, pp. 1942–1948 (1995).
63. Otero, F.E.B.; Freitas, A.A.; Johnson, C.G.: A new sequential covering strategy for inducing classification rules with ant colony algorithms. *IEEE Trans. Evol. Comput.* **17**(1), 64–76 (2012)
64. Yang, L.; Li, K.; Zhang, W.; Ke, Z.: Ant colony classification mining algorithm based on pheromone attraction and exclusion. *Soft. Comput.* **21**(19), 5741–5753 (2017)
65. El-Affendi, M.A.; Alrajhi, K.; Hussain, A.: A novel deep learning-based multilevel parallel attention neural (MPAN) model for multidomain arabic sentiment analysis. *IEEE Access* **9**, 7508–7518 (2021)
66. Brahimi, B.; Touahria, M.; Tari, A.: Improving sentiment analysis in Arabic: A combined approach. *J. King Saud. Univ. Inf. Sci.* **33**(10), 1242–1250 (2021)
67. Moudjari, L.; Benamara, F.; Akli-Astouati, K.: Multi-level embeddings for processing Arabic social media contents. *Comput. Speech Lang.* **70**, 101240 (2021)
68. Abd Elminaam, D.S.; Neggaz, N.; Ahmed, I.A.; Abouelyazed, A.E.S.: 'Swarming behavior of Harris hawks optimizer for Arabic opinion mining. *Comput. Mater. Contin.* **69**(3), 4129–4149 (2021)
69. Tubishat, M.; Abushariah, M.A.M.; Idris, N.; Aljarah, I.: Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. *Appl. Intell.* **49**(5), 1688–1707 (2019). <https://doi.org/10.1007/s10489-018-1334-8>
70. Atia, S.; Shaalan, K.: Increasing the accuracy of opinion mining in Arabic. *Arab Comput. Linguist.* (2015). <https://doi.org/10.1109/ACLing.2015.22>
71. El-Defrawy, M.; El-Sonbaty, Y.; Belal, N.A.: A rule-based subject-correlated Arabic stemmer. *Arab J. Sci. Eng.* **41**(8), 2883–2891 (2016)
72. Agarwal, B.; Mittal, N.: *Prominent Feature Extraction for Sentiment Analysis*. Springer, Berlin (2016)
73. Pang, B.; Lee, L.; Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86 (2002).
74. Pang, B.; Lee, L.: Opinion mining and sentiment analysis. *Found. Trends@ Inf. Retrieval*, 2(1–2), 1-135 (2008)
75. Agarwal, B.; Mittal, N.: Text classification using machine learning methods-a survey. In: *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012)*, December 28–30, 2014, pp. 701–709 (2012).
76. Zomorodi-moghadam, M.; Abdar, M.; Davarzani, Z.; Zhou, X.; Pławiak, P.; Acharya, U.R.: Hybrid particle swarm optimization for rule discovery in the diagnosis of coronary artery disease. *Expert Syst.* **38**(1), e12485 (2021)
77. Casillas, J.; Carse, B.; Bull, L.: Fuzzy-XCS: a Michigan genetic fuzzy system. *IEEE Trans. Fuzzy Syst.* **15**(4), 536–550 (2007)
78. Mahmoudi, M.R.; Behboodian, J.; Maleki, M.: Large sample inference about the ratio of means in two independent populations. *J. Stat. Theory Appl.* **16**(3), 366–374 (2017)
79. Haghbin, H.; Mahmoudi, M.R.; Shishebor, Z.: Large sample inference on the ratio of two independent binomial proportions. *J. Math Ext.* (2015).
80. Frank, E.; Witten, I. H.: Generating accurate rule sets without global optimization. In: *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 144–151 (1998).
81. Cohen, W.W.: Fast effective rule induction, in *Machine learning proceedings*. Elsevier **1995**, 115–123 (1995)
82. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. *Mach. Learn.* **11**(1), 63–90 (1993)
83. Quinlan, J.R.: Simplifying decision trees. *Int. J. Man Mach. Stud.* **27**(3), 221–234 (1987)
84. *Data Mining Software in Java*, 2011. <http://www.cs.waikato.ac.nz/ml/weka>.
85. Cortes, C.; Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995). <https://doi.org/10.1023/A:1022627411411>
86. Sebastiani, F.: Machine learning in automated text categorization. *ACM Comput. Surv.* **34**(1), 1–47 (2002)
87. Murphy, K. P.: *Naive Bayes classifiers*. University of British Columbia (2006)
88. Abbasi, A.R.; Mahmoudi, M.R.; Avazzadeh, Z.: Diagnosis and clustering of power transformer winding fault types by cross-correlation and clustering analysis of FRA results. *IET Gen. Trans. Distrib.* **12**(19), 4301–4309 (2018)
89. Mahmoudi, M.R.; Nematollahi, A.R.: On the detection and estimation of the simple harmonizable processes. *Iran J. Sci. Technol.* **39**(2), 239–242 (2015)
90. Abbasi, A.R.; Mahmoudi, M.R.; Arefi, M.M.: Transformer winding faults detection based on time series analysis. *IEEE Trans. Instrum. Meas.* **70**, 1–10 (2021)
91. Zarei, A.R.; Shabani, A.; Mahmoudi, M.R.: Comparison of the climate indices based on the relationship between yield loss of rain-fed winter wheat and changes of climate indices using GEE model. *Sci. Total Environ.* **661**, 711–722 (2019)
92. Hossain, S.K.M.; Ema, S.A.; Sohn, H.: Rule-based classification based on ant colony optimization: a comprehensive review. *Appl. Comput. Intell. Soft. Comput.* 2022 (2022).

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.