IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Political Arabic Articles Orientation Using Rough Set Theory with Sentiment Lexicon

Jwan K. Alwan [1], Abir Jaafar Hussain[2], Dhafar Hamed Abd[3], Ahmed T. Sadiq[4], Mohamed Khalaf[3] and Panos Liatsis[5]

[1]Biomedical Informatics College, University of Information Technology and Communications, Baghdad, Iraq
[2]Department of Computer, Liverpool John Moores University, Liverpool L3 3AF, U.K
[3]Department of Computer, Al-Maarif University College, Al-Anbar, Iraq
[4]Department of Computer, University of Technology, Baghdad, Iraq
[5]Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, UAE

Corresponding author: Jwan K. Alwan (jwanism@uoitc.edu.iq)

ABSTRACT Sentiment analysis is an emerging research field that can be integrated with other domains, including data mining, natural language processing and machine learning. In political articles, it is difficult to understand and summarise the state or overall views due to the diversity and size of social media information. A number of studies were conducted in the area of sentiment analysis, especially using English texts, while Arabic language received less attention in the literature. In this study, we propose a detection model for political orientation articles in the Arabic language. We introduce the key assumptions of the model, present and discuss the obtained results, and highlight the issues that still need to be explored to further our understanding of subjective sentences. The main purpose of applying this new approach based on Rough Set (RS) theory is to increase the accuracy of the models in recognizing the orientation of the articles. We present extensive simulation results, which demonstrate the superiority of the proposed model over other algorithms. It is shown that the performance of the proposed approach significantly improves by adding discriminating features. To summarize, the proposed approach demonstrates an accuracy of 85.483%, when evaluating the orientation of political Arabic datasets, compared to 72.58% and 64.516% for the Support Vector Machines and Naïve Bayes methods, respectively.

INDEX TERMS Arabic political article, Support vector machines, Naïve Bayes, Rough set theory, N-gram, Sentiment lexicon

I. INTRODUCTION

The prevalence of online social networks, ideological websites and newspapers has led to increased research interest in fields concerned with the analysis of such resources to extract useful information. Sentiment analysis (SA) (also known as opinion mining or sentiment orientation) is concerned with the identification of sentiment orientation from formless data [1]. It can be thought of as a classification activity, which decides the sentiment or view carried in a specific sentence or article as being negative, positive, or neutral. The majority of the researches are conducted in the English language, while research in the Arabic language is still at its infancy. However, the Arabic language is different from English and possesses its own issues as well as challenges.

The Arabic language is considered as the fifth most spoken language in the world. According to the latest statistics, there are more than 422 million people speaking Arabic as a first language and around 250 million people, which speak Arabic as their second language [3]. The Arabic alphabet includes 28 letters. Arabic letters do not have upper or lower case. In terms of orientation, Arabic is written from right to left [4]. There is a limited amount of research on sentiments, attitudes, emotions and opinions in the context of Arabic. The vast majority of previous studies focused on specific article categories, for instance, political articles to identify and categorize the political class, and sport articles to classify the supported teams. In this study, we attempt to apply orientation recognition of polarity, by gathering and analysing a dataset pertaining to Arabic political articles

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3054919, IEEE Access

based on social networks, ideological websites and newspapers.

There are three main classification levels in SA, i.e., document-level, sentence-level, and aspect-level. This paper focuses on document-level analysis, aiming to classify an opinion article as being positive or negative. The entire article is considered as the basic information unit (since it normally focuses on discussing a single topic). The aim of the current work is to analyse articles to determine their political orientation. The decision will rely on Rough Set (RS) theoryand the sentiment lexicon of the studied texts. The specific political orientations considered in this work are Reformist, Conservative and Revolutionary.

This study is divided into three main parts. Firstly, we will introduce the development of a manually annotated corpus of political articles written in Arabic. In this context, we will present relevant statistical features of the corpus, such as the total numbers of articles, sentences, words, punctuations, unique words, words per article, and sentences per article. Secondly, we will develop different configurations of the feature vector for both the lexicon- and machine learning-based approaches. Thirdly, we will present our experiments performed on a hybrid (combination of lexicon and RS theory) representation and machine learning approach for polarity detection as well as categorisation.

Lexicon contains a list of words or phrases and it is an important resource in sentiment analysis [1, 2]. There have been a multitude of approaches for both manual [3-5] and automated [6-8] lexicon construction. The former approach is time consuming and does not work in a variety of domains, while the latter has become a hot research topic since it is easy to use and works in any domain [9].

A number of methods focused on sentiment polarity word detection, using lexicon construction [10-13]. However, they possess certain limitations. First, some of the methods use manual lexicon definition [11, 12], which makes their application in other domains cumbersome. Second, the lexicon approach does not always provided high accuracy and performance [14, 15]. In order to address this limitation, this research proposes a methodology for automated lexicon construction, where RS theory is used to solve the problem of low accuracy. The results of our work are benchmarked with state-of-the-art machine learning (ML) algorithms, i.e., Support Vector Machines (SVM) and Naïve Bayes (NB).

Pawlak [16, 17] used RS theory as an efficient mathematical tool to handle uncertain, inexact or vague information, which garnered the interest of numerous researchers to contribute towards its applications and further development [18-25]. In data analysis, the key benefit pertaining to Rough Set theory is the lack of requirements for any additional or preliminary information, regarding the data, for instance, basic probability assignments pertaining to the Dempster–Shafer theory, probability distributions associated with the use of statistics or the degree of membership related to Fuzzy Set theory.

When considering the state-of-the-art in the field of the current research, published works neither consider different aggregation methods, nor perform comparisons with ML algorithms. To summarize, we propose a novel methodology for document sentiment representation and analysis, which is systematically evaluated and compared with appropriate ML algorithms, namely, SVM and NB, which have been shown to successfully handle text classification tasks [26, 27].

The remainder of this paper is organised as follows. Section II presents an overview of related works. In section III, the proposed system solution is introduced. Section IV presents the outcomes of the experiments and their analyses. The conclusions and suggestions for future research are made in section V.

II. BACKGROUND

In this section, the models used in our experiments will be presented including Rough Set theory, Support Vector Machines and Naïve Bayes algorithms.

A. Rough Set theory

Pawlak (1982) used Rough Set theory as a new intelligent mathematical tool to handle incompleteness as well as uncertainty. RS theory considers the issues pertaining to a lower as well as an upper approximation of a set, and associated models of sets and the approximation space. A key application related to RS theory is attribute reduction, i.e., elimination of least informative attributes. Attribute reduction can be achieved, when equivalence relations are produced by comparing sets of attributes. By employing the dependency degree as a measure, removal of attributes is performed and the reduced attribute set offers an identical degree of dependency to the original set. In this section, we introduce key concepts of RS theory to facilitate the analysis performed in this work. A full treatment of RS theory and its concepts can be found in (Pawlak, 1982, 1996).

An Information System is employed to represent knowledge in rough sets, which is given as a 4-tuple $IS = <U, A, V, f>$, where U denotes the closed universe, a finite set pertaining to N objects $\{x_1, x_2, \ldots, x_n\}$, and A refers to a finite set of attributes $\{a_1, a_2, \ldots, a_n\}$ that can be further segmented into two disjoint subsets, i.e., C and D, $A = \{C \cup D\}$, where C defines the condition attributes and D signifies a set of decision attributes. Set $V = \cup_{a \in A} V_a$, where V_a refers to a domain of the attribute a, while $f: U \times A \rightarrow V$ relates to the total decision function, referred to as the information function, wherein $f(x, a) \in V_a$ for each $a \in A, X \in U$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3054919, IEEE Access

In RS theory, the upper and lower approximations are regarded as two basic operations, related to any concept X ⊆ *U*, while attribute set R ⊆ A, and *X* can also be approximated via the upper and lower approximations. The lower approximation pertaining to *X* can be defined as a set of objects of *U* that are certainly in *X*, represented as:

$$\underline{R}(X) = \{x \in U: [x]_R \subseteq X\} \quad (1)$$

The upper approximation pertaining to *X* can be defined as a set of objects of *U* that could probably be in *X*, represented as:

$$\overline{R}(X) = \{x \in U: [x]_R \cap X \neq \emptyset\} \quad (2)$$

B. Support Vector Machines

Support Vector Machines are widely used and popularly employed as classifiers [28]. The basic concept behind SVM is to employ hyperplanes to separate dissimilar classes. On the other hand, SVM has been much admired for its precision, when handling linearly separable data, while its performance falls short when dealing with non-linearly separable data [29]. This limitation can be addressed through the use of the so-called kernel trick, which allows the representation of the problem into a higher dimensional space, typically associated with linear separability or a reduction in the non-linearity of the problem.

The key concept behind SVM is to choose an appropriate kernel function, and to appropriately adjust the kernel parameters [30]. With regards to computational complexity, determining the most appropriate decision plane is posed as a quadratic optimisation problem. An appropriate decision hyperplane could allow the generation of robust class decisions based on the kernel function, via the nonlinear transformation, as represented in Eq. (3):

$$D(x) = w^* \varphi(x) + b = \sum_{i=0}^{N} y_k a_i^* (K(x_i, x) + b) \quad (3)$$

w^* denotes the weight vector that specifies the hyperplane with maximum margin, $\varphi(x)$ are the predefined functions of input vector *x*, a_i^* corresponds to the optimal coefficients as determined during training, *K()* is the kernel function, *y* indicates the class labels (target outputs), and the parameter *b* is the bias.

C. Naive Bayes

Since the 1960s, the Naïve Bayes classifier has been widely studied. In the early 1960s, it was introduced (under a different name) in the text retrieval community, and continues to be a popular method to categorise text, i.e., determining the categories of documents (e.g., politics or sports, legitimate or spam, etc.) with word frequencies being used as the features. In [31], the researchers employed a model by considering the conditional probabilities.

For each article, the features or words pertaining to the article were calculated [32], by employing Eq. (4) as :

$$y = P(c_k) \prod_{i=1}^{n} P(x_i|c_k) \quad (4)$$

In this case, c_k relates to the class labels, x_i is a d-dimensional feature vector, and *y* denotes the final class for the text, which is used to compare with the labels of the data. The researchers employed Eq. (3) to calculate the minimum or maximum *y* by accounting for the use of NB. They also determined the values pertaining to $P(c_k)$ and $P(x_i|c_k)$ by employing the two formulae shown in Eq. (5), since $P(c_k)$ is identical for all NB, while the value of $P(x_i|c_k)$ varies, based on the classifier employed:

$$P(c_k) = \frac{document P(c=c_k)}{N_{doc}}$$
$$P(x_i|c_k) = \frac{count\ (x_i|c_k)}{\sum_{x \in v}\ count(x|c_k)} \quad (5)$$

d. Polarity of the Arabic Articles

This section focuses on reviewing several Arabic text polarities or orientation methods. Techniques used in these studies for opinion mining are investigated. The limitations of each technique will be also discussed.

Hmeidi et al., [45] tested five classifiers (SVM, NB, KNN, Decision Tree, and Decision Table) with three different versions of datasets (preprocessing, light10 stemmer and Khoja stemmer). They evaluated the accuracy and scalability of two popular machine learning tools (i.e., Weka and RapidMiner) to examine their advantages and disadvantages for Arabic text categorization (TC). The experimental results revealed that SVM with the light10 stemmer gives the best results amongst the considered classifiers, outperforming the root-based stemmer in terms of accuracy. Finally, they recommended the use of RapidMiner due to its efficiency and scalability to Arabic TC researchers.

Al-Radaideh and Al-Khateeb [46] applied the associative classifier method to classify Arabic articles related to the medical domain. The experimental results reported by the authors showed that the associative classification method outperformed the C4.5, Ripper, and SVM algorithms based on a corpus of 1000 Arabic medical articles that belong to 10 different diseases (classes).

Rbooraig et al., [47] proposed a new method for automatic categorization of Arabic articles based on political orientation. The method starts by collecting texts for building a corpus, then proceeds to examine the performance of various feature reduction approaches. They utilized two popular feature extraction techniques, i.e., traditional text categorization and stylometric features (SF). Their worked considered six algorithms, namely, NB, Discriminative Binomial NB (DMNB), Sparse Generative Model (SGM)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3054919, IEEE Access

classifier, SVM, Random Forest (RF), and ensembles of classifiers (VOTE). The highest accuracy was obtained when using TC.

Al-Radaideh et al., [48] proposed a new method for Arabic text categorization using term weighting and multiple reductions. This uses term weight to extract weights from the text, followed by the use of RS theory to reduce the number of terms used in generation of classification rules. A quick reduction algorithm was used, while multiple reductions were used to generate the set of classification rules that represent the RS classifier. An Arabic corpus, consisting of 2700 documents belonging to nine categories, was used to evaluate performance. The experimental results revealed that the method achieved higher accuracy compared to k-nearest neighbour and decision tree (DT) algorithms.

Contrary to the directions pursued in the literature, in this work, we focus on analysing a dataset collected from political Arabic articles, and useRS theory and machine learning methods to determine their orientation and evaluate performance.

III. METHODOLOGY

Recently, mining through web content related to business reviews, particularly for poorly represented languages, such as Arabic, attracted the interest of many researchers [33]. In our work, we built a manually annotated corpus of political articles, written in Arabic. Next, we created different versions of the feature vector for both lexicon-based and machine learning approaches. Finally, we experimented on a hybrid representation (i.e., combination of lexicon and rough set theory), coupled with machine learning for orientation recognition of polarity. Figure 1 shows the proposed hybrid model based on Lexicon and Rough Set theory (LRST).

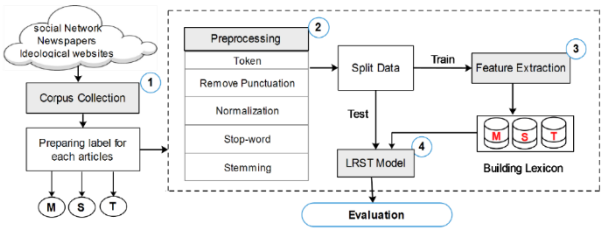

Figure 1: An overview of the proposed LRST model. The encoding labels are S for reformist, M for conservative and T for revolutionary.

As illustrated in Figure 1, the first step is to provide a labelled dataset for training and testing purposes. The next step is feature extraction. Once the features are computed, the dataset is split into training and testing sets. The former is fed into a classification algorithm, whose performance is evaluated on the testing set.

A. Corpus Description and Preparation

Determining the orientation of opinions from text is a sub-field of machine learning, which aims at understanding the context of the words. This requires access to large amounts of domain-specific benchmark datasets that are collectively used to train sentiment classifiers. These datasets may include imbalanced as well as balanced data. The Political Arabic dataset includes 206 Arabic documents with different lengths, which can be segmented in 3 categories, labelled as Reformist, Conservative and Revolutionary, respectively. Using this dataset, the effectiveness of the proposed method can be evaluated. The corpus is published on the Mendeley data [34].

TABLE 1: DESCRIPTION OF CORPUS

Document number	Label of document in Arabic	Encoding label
80	تيار اصلاحي (Reformist Party)	S
58	تيار محافظ (Conservative Party)	M
68	تيار ثوري (Revolutionary Party)	T

TABLE 2: TEXT STATISTICS FOR EACH CORPUS CLASS

Statistics	Conservative	Reform	Revolutionary	Total
Number of articles	58	80	68	206
Number of sentences	421	652	762	2122
Highest sentence articles	53	40	56	149
Lowest sentence articles	1	3	1	5
Number of all words	14111	29607	23853	67571
Highest word articles	626	1130	1208	2964
Lowest word articles	54	82	63	199
Number of unique tokens	3246	6782	5303	15331
Tokens occurring more than one time	1997	3319	2646	7962
Number of English words	65	110	7	182
Number of punctuations	616	1271	1054	2941
Number of digits	88	207	211	506

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3054919, IEEE Access

Data preparation can be defined as the process wherein raw data is labelled. In this case, a label is made for every article. When data is gathered from various platforms, such as social networks, ideological websites and newspapers, the data is typically unlabelled and hence data annotation is required. First, the data needs to be collected, and the text associated to an article needs to be saved in a text file. For example, if 100 articles are collected, and each article is saved in text file form, there will be 100 text files. Second, a folder is created, wherein all the text files belonging to that folder are gathered. For example, if 100 articles have been collected related to two classes, then two folders are created, based on the class labels, and the associated text files are placed in the corresponding folder. Third, Python was used to implement the program that creates the Excel file, and to label each article according to its class. After this process is completed, the Excel file can be used in the following stages.

B. Arabic Language Pre-processing

This stage is crucial in making accurate decisions efficiently. A number of steps are necessary, including handling punctuations, tokenisation, stop-words, normalisation and stemming. Tokenisation is the first step to tokenise a document to words, which is then sent to the next step, i.e., punctuation to remove Arabic punctuations by employing a token length of less than 2.

In Arabic language, normalisation is crucial in forming words in a specific writing style. In this paper, we employed the following processes. The first step was to remove diacritics such as ً,ٌ,َ,ُ,ِ,ّ,ـ,ْ,ٍ,ٰ. The second step was to remove elongated letters, since they create issues, as the same word could appear longer than four times, e.g., with longer flat line parts (ـــ). Therefore, the feature could be large and the decision could become incorrect. In this step, "tatwel" was removed. For example, if we consider the set of (ثـــــورة,ثـــورة, ثـورة,ثورة), although all of these words are identical, with tatwel, they will appear different and hence the aim of this step was to formalize the appearance of Arabic words (ثورة). The final step was to substitute variations in the use of the letters of the alphabet by following a simple process. For instance, the letter "Alif" (أ,إ, آ,ٱ, ا) is normalised to letter "Alif" (ا), letter Alif-Maqsura (ى) is normalised to letter "Ya" (ي), and letters such as "Waw" and "Ya" (ؤ,ئ) would become the letter "Hamzah" (ء), and the letter "Ta-Marbota" (ة) would become normalised to the letter "Ha" (ه).

The next step was to identify and remove stop-words so as to improve the compactness of information in the feature vector. In this way, the feature vector will only contain important information related to polarity orientation. In this case, two types of stop-words were employed, i.e., sundered Arabic stop-words such as (على,الى , من, في , ذلك, …), while the second type of stop-words has no effect.

Stemming is also crucial in reducing the size of the feature vector [35]. In this case, certain words are removed since they are stop-words, or relate to a word already existing in the feature vector. Stemming includes two types [26], i.e., root stem [36] and light stem [37, 38]. In this research, we employed light stemming using the Information Science Research Institute's (ISRI) stemmer [39].

C. Feature Extraction Using Machine Learning Algorithms

In machine learning-based text classification, the first core step is the transformation of the text into a numerical representation, which is typically represented by a feature vector. This process is also referred as text vectorisation or feature extraction. In this work, we applied two methods, namely, the bag-of-words (i.e., term frequency (TF)) and the n-gram approaches. This type of representation allows similar words to be mapped to similar feature representations, therefore enhancing the classifier's performance.

The n-gram approach has been suggested as a means of representing textual characteristics by Sanderson and Guenther [40] and Peng et al., [41]. N-gram for sentiment analysis has also been employed in [2]. There are various sizes of n-grams [42-44] i.e., 1, 2, 3, 4 and 5. When a size of 1 is used, the corresponding representation is known as a unigram, while a size of 2 is known as bigram, a size of 3 is known as a trigram, and 4 and 5 correspond to 4-gram and 5-gram, respectively. In this work, we employed a variety of n-gram sizes, specifically, unigram, bigram, trigram, 4-gram and 5-gram, to validate the proposed model and associated machine learning algorithms.

Term Frequency (TF) refers to the frequency of occurrence related to a particular term. As there are different lengths for each article, it is not unusual for a particular term to be repeated more often in longer articles compared to shorter articles. In order to incorporate robustness in regards to the size of the article, we implemented simple normalisation, where the term frequency is found by dividing the number of times a particular term appears by the total number of terms in the document:

$$TF = \frac{n}{\sum T_n} \tag{6}$$

where n represents the occurrence pertaining to a term in the article, and T_n denotes the overall count of all the terms in the article.

D. Feature Extraction Using the Hybrid LRST Model

Next, the lexicon for each class was constructed and used in the proposed LRST model. Assume m to be the number of articles, n the number of labels and w to be the number of words in the article. Let X to be the collection of articles $X = \{A_1, \dots, A_i, \dots A_n\}, where\ A_i\ is\ an\ article,\ i \in \mathbb{Z}^+$ and C is the collection of labels for the articles, $C = \{l_1, \dots, l_j, \dots l_n\}, where\ l_j\ is\ the\ label\ of\ the\ article,\ j \in$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3054919, IEEE Access

$\mathbb{Z}^+$. *C* makes a partition on *X* such that $A_i \in l_j$ for the same *j*, when $A_i \in l_j$, we refer to A_i by $A_{i,j}$.
Assume *V* to be the lexicon, where *V* is constructed for each article in dataset *X* with labels from set *C* as shown:

$$\begin{aligned} P_1 &= \cup A_{i,1} |\ A_{i,1} \in l_1 \\ P_2 &= \cup A_{i,2} |\ A_{i,2} \in l_2 \\ P_3 &= \cup A_{i,3} |\ A_{i,3} \in l_3 \end{aligned} \quad (7)$$

Eq. 7 makes a partition such that each article must belong to exactly one partition, i.e., the corresponding partitions are disjoint. In this research, three unique labels are considered, i.e., reformist, conservative and revolutionary, corresponding to each of the partitions P_1, P_2 and P_3. In general, P_j is given as:

$$P_j = \bigcup \{ A_{i,j} \mid 1 \le i \le m, 1 \le j \le 3 \} \quad (8)$$

where *i* is the number of articles that belong to class *j*. The lexicon, i.e., set *V*, is constructed for each class as shown:

$$V_j = \{ w \mid w \in P_j, 1 \le j \le 3 \} \quad (9)$$

E. Feature Extraction

Formally, an orientation polarity system can be regarded as a system *PS = (V, T)*, where *V* is the universe, i.e., a finite set of feature vectors, trained on words we already built and called the lexicon. *T* is a non-empty finite set of words in the article that would be used to test and determine the orientation of article. *T* contains a set of words $T = \{w_1, w_2, w_3, \ldots, w_r\}$, where *r* is the numbers of words in the article. Building *V* and test article *T* depends on 5-gram numbers.
From RS theory, we make use of the lower approximation because there is intersection between V_j and *T*. Thus, since $w \subseteq T$, i.e., each word *w* belongs to *T*, we calculate $\underline{B}(w)_j$, which represents the number of matches:

$$\underline{B}(w)_j = \{ w \mid w \in T \text{ and } w \in V_j \} \quad (10)$$

where $1 \le j \le 3$ counts the number of classes in lexicon V_j. In order to test an article *T*, it is important to perform the test in each class of V_j, and then consider the numbers of class matches in the article. In this case, we use Eq. 11 to determine:

$$Pr = argmax\left(\sum_{j=1}^{n} \underline{B}(w)_j\right) \quad (11)$$

where *Pr* the predicted class for article *T*.

G. Evaluation Metrics

To analyse the performance of the proposed approach, we use a number of well-known classification metrics, i.e., accuracy, recall, precision, F1-score, and Kappa, as shown in Table 3. These metrics involve parameters such as true positives (TP), true negatives (TN), false positives (FP), false negatives (FN), actual observed agreement (P_o), and chance agreement (P_e).

.

TABLE 3: EVALUATION METRICS

Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Recall	$\frac{TP}{TP + FN}$
Precision	$\frac{TP}{TP + FP}$
F1-score	$2 * \frac{Recall * Precision}{Recall + Precision}$
Kappa	$k = \frac{P_o - P_e}{1 - P_e}$

Precision refers to the ratio of accurately estimated positive observations against the overall predicted positive observations. On the other hand, recall refers to the ratio of accurately estimated positive observations against the overall observations in the actual positive class. The weighted average of precision and recall is defined as the F1-score. Hence, this metric takes into consideration both the false negatives and false positives. It is quite difficult to have an intuitive understanding of accuracy. However, the F1-score is generally more beneficial compared to accuracy, particularly when the dataset is characterised by a non-uniform class distribution, i.e., class imbalance. Accuracy works well when false negatives and false positives have comparable costs. However, when the cost of false negatives and false positives is weighed differently, it is recommended to consider both recall and precision.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Our corpus was split to 70% and 30% for training and testing, respectively, as illustrated in Table 4.

TABLE 4: NUMBER OF TRAINING AND TESTING SAMPLES FOR EACH CLASS

Class	Training patterns	Testing patterns
M	43	15
S	46	22
T	55	25
Total	144	62

To analyse the efficiency of the proposed approach, experiments were conducted on the three groups or

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3054919, IEEE Access

categories. The LRST algorithm was applied to select major characteristic subsets. Table 5 indicates the number of each class present in the feature vector.

Unique words for all class will be used for TF as shown in Eq. 6, since it can have one feature matrix for all classes with no similarity. It should be noted that only TF was used rather than TF-IDF (Term Frequency- Inverse Document Frequency), since we are investigating words within the same article, and hence the feature vector utilized for the classifier will contain words from the same article. TF-IDF determines the relation between article and corpus, which is not needed in the context of our investigations. The proposed model has three vectors because we have three classes. As it can be seen in Table 5, the total numbers with and without similarity are close in the case of the 3-gram, but also for 4-gram and 5-gram representations.

TABLE 5: NUMBERS OF FEATURES EXTRACTED USING N-GRAM

Class	**Numbers of each class**				
	1-gram	2-gram	3-gram	4-gram	5-gram
M	1593	3527	3720	3711	3679
S	3992	10047	10852	10946	10957
T	3252	7009	7503	7547	7546
Total with similarity	6142	20197	22056	22204	22182
Total Without Similarity	8837	20583	22075	22204	22182

As the proposed method does not use numerical values, the n-gram approach is chosen. However, the selected machine learning algorithms, i.e., NB and SVM, require numerical features, and thus, TF is applied using the n-gram inputs.

TABLE 6: NUMBER OF CORRECT LABELS FOR THE PROPOSED METHOD COMPARED TO MACHINE LEARNING APPROACHES

Algorithm	Label	1-gram	2-gram	3-gram	4-gram	5-gram
LRST	M	6	10	7	8	7
	S	25	22	15	8	7
	T	18	21	16	12	9
SVM	M	3	0	0	0	0
	S	25	25	25	25	25
	T	17	0	0	0	0
NB	M	15	8	7	7	7
	S	13	19	17	17	17
	T	12	6	5	4	3

Figure 2 shows the results for the proposed method, benchmarked against SVM and NB for each n-gram representation (n=1,…,5). As shown in the Figure, the proposed method displays superior performance when using the 1-gram, 2-gram and 3-gram representations. For the 4-gram case, LRST has similar performance to NB, but better than SVM. Finally, for the 5-gram case, the proposed model has lower performance than the machine learning methods.

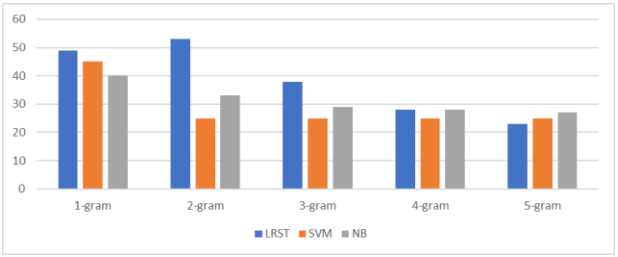

Figure 2: Correct labels for SVM, NB, and LRST.

As mentioned previously, the main contribution of this research is to recognise geometrical configurations with classes using text mining. In the proposed model, we applied SA using various parameter configurations. The results illustrated in Tables 7 and 8 indicate that when using 1-gram and 2-gram, SA yields better accuracy, i.e., accuracies of 79.032% and 85.483% were obtained, for the 1-gram and 2-gram representations, respectively.

TABLE 7: EVALUATION PERFORMANCE METRICS BASED ON 1-GRAM

Method	Precision	Recall	F1-score	Kappa	Accuracy
LRST	0.85	**0.79**	**0.77**	**0.665**	**79.032%**
SVM	0.84	0.73	0.69	0.556	72.580%
NB	**0.86**	0.65	0.67	0.494	64.516%

TABLE 8: EVALUATION PERFORMANCE METRICS BASED ON 2-GRAM

Method	Precision	Recall	F1-score	Kappa	Accuracy
LRST	**0.89**	**0.85**	**0.87**	**0.778**	**85.483%**
SVM	0.16	0.40	0.23	0.0	40.322%
NB	0.68	0.53	0.53	0.300	53.225%

TABLE 9: EVALUATION PERFORMANCE METRICS BASED ON 3-GRAM

Method	Precision	Recall	F-score	Kappa	Accuracy
LRST	**0.91**	**0.61**	**0.73**	**0.490**	**61.290%**
SVM	0.16	0.40	0.23	0.0	40.322%
NB	0.64	0.47	0.46	0.200	46.774%

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3054919, IEEE Access

The decision in this model presents a trade-off, since the collected datasets were limited by the number of documents. We performed extensive simulation experiments, which showed that in the majority of cases, the NB and SVM classifiers yield lower results based on the performance metrics in comparison to LRST. As shown in Tables 9 and 10, the proposed LRST model performed well in comparison to NB and SVM with an accuracy of 61.290% with the 3-gram representation.

TABLE 10: EVALUATION PERFORMANCE METRICS BASED ON 4-GRAM

Method	Precision	Recall	F1-score	Kappa	Accuracy
LRST	**1**	**0.45**	**0.62**	**0.351**	**45.161%**
SVM	0.16	0.40	0.23	0.0	40.322%
NB	0.63	**0.45**	0.43	0.178	**45.161%**

The proposed method based on 5-gram achieved a performance of 37.096 %, contrary to SVM, which obtained an accuracy of 40.322%, and NB which was the top performer with an accuracy of 43.548 %, as shown in Table 11. Our empirical study provides evidence to support the robust performance of LRST, compared to the use of machine learning methods. Overall, the results demonstrate good potential for polarity orientation detection in the context of text mining classification. This is also illustrated by the Kappa values, as shown in Tables 7-9. Clearly, in order to obtain satisfactory results, it is vital to select the appropriate model. The Naïve Bayes classifier provided a good performance in the text mining datasets; however, this was still not at the optimal level. The best results in terms of performance metrics were obtained for the 2-gram representation, as shown in Figure 3.

TABLE 11: EVALUATION PERFORMANCE METRICS BASED ON 5-GRAM

Method	Precision	Recall	F-score	Kappa	Accuracy
LRST	**0.96**	0.37	**0.53**	**0.276**	37.096%
SVM	0.16	0.40	0.23	0.0	40.322%
NB	0.63	**0.44**	0.40	0.153	**43.548%**

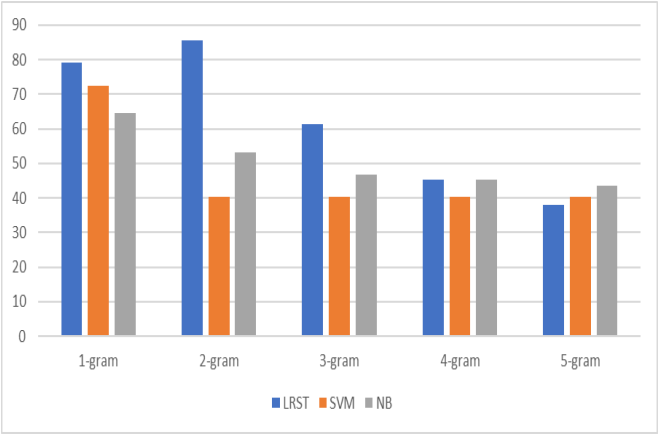

Figure 3: Model accuracy versus n-gram representation.

The purpose of the first experiment was to compare the performance of the LRST method with the SVM and NB algorithms. Tables 7, 8, 9, 10 and 11 show the precision, recall, F1-scores, kappa and accuracy values using various N-gram representations. As illustrated in Fig. 2, the proposed method has higher accuracy, when compared to SVM and NB. Tables 9, 10 and 11 indicate that the accuracy of the proposed method using 3-gram, 4-gram and 5-gram representations, is significantly lower than in the case of 1- and 2-gram representations. To verify the accuracy of the proposed model, it was compared to NB and SVM on the corpus. As shown in Table 10, LRST shows higher accuracy than SVM and similar accuracy to the NB method. Generally, compared with the SVM and NB, LRST achieved lower classification error rates. Based on these observations, it was found that the proposed model attained the best results (i.e., an accuracy of 85.483%) with the 2-gram representation, which is substantially higher than the results of the ML techniques. Indeed, this improvement in performance is robust to the type and size of the feature representation. For instance, in the case of 1-, 2-, 3-, 4-grams, LRST showed better performance than SVM and NB, except for the case of 5-gram, where SVM and NB showed better accuracy.

To investigate the generalization of the proposed method, the BBC news articles dataset was used [49], available from UCD[1]. This dataset is written in the English language, collected from BBC news websites in the period between 2004 and 2005. The BBC dataset contains five classes, as illustrated in

TABL 12. A total of 225 records (articles) are contained in this dataset, which is much higher that the political Arabic articles in the first set of experiments. In this dataset, we applied the lower approximation method with a lexicon-vector.

TABLE 12: TRAINING AND TESTING SAMPLES FOR THE BBC DATASET

Class	Training	Testing	Total
Business	368	142	510
Entertainment	274	112	336
Politics	276	141	417
Sport	359	152	511
Tech	280	121	401
Total	1557	668	2225

TABLE 13: ACCURACY OF LRST METHOD FOR UNIGRAM REPRESENTATION

Class	Precision	Recall	F1-	Accuracy

[1] mlg.ucd.ie/datasets/bbc.html

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3054919, IEEE Access

			score	
Business	0.95	0.95	0.95	
Entertainment	**1.00**	0.94	0.97	
Politics	0.96	0.97	0.96	96.706%
Sport	0.99	0.98	**0.99**	
Tech	0.94	**0.99**	0.97	

Table 13 shows the results of applying the proposed method with lexicon on the raw dataset. The results show an accuracy of 96.706% for all classes. The achieved value for recall is 0.99 in class Tech, precision for the Entertainment class is 1.00, and F1-score for the Sport data is 0.99. In general, the achieved accuracy is consistent over all classes and supports the robustness of the LRST method. In this experiment, the unigram representation was utilised since we were interested in the ability of the proposed method to deal with different corpora, using another language, and with a higher number of articles.

V. CONCLUSIONS

Sentiment analysis has been proved to be a difficult field to explore as it poses various impediments related to natural language processing. It offers an extensive range of applications, which could benefit from its use. Some of these applications include marketing, news analytics, and etc. An important aspect of the research described in this contribution is that it targets documents written in Arabic. Specifically, a new approach was proposed for Arabic language articles based on RS theory. The context of the particular application was sentiment analysis of political articles. Various pre-processing techniques were used to standardize the presentation of textual information, and in effect, minimise inherent noise. Furthermore, lexicon was used to signify the extent of negativity or positivity of every term presented in the lexicon. Our study indicated that rough set theory offers improved results when using sets of documents as input for the analysis of sentiment in comparison to state-of-the-art machine learning algorithms. To demonstrate the potential use of the proposed approach, we considered the BBC news articles dataset, which contains a larger number of documents, in the English language, representing five different document categories.

Future work will involve the use of the LRST method in extracting information from video feedback used by companies to gain details about their products using text mining and natural language processing.

REFERENCES

[1] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining text data*: Springer, 2012, pp. 415-463.

[2] A. Dey, M. Jenamani, and J. J. Thakkar, "Senti-N-Gram: An n-gram lexicon for sentiment analysis," *Expert Systems with Applications*, vol. 103, pp. 92-105, 2018.

[3] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, 2005, pp. 347-354.

[4] T. I. Jain and D. Nemade, "Recognizing contextual polarity in phrase-level sentiment analysis," *International Journal of Computer Applications*, vol. 7, no. 5, pp. 12-21, 2010.

[5] S. Huang, Z. Niu, and C. Shi, "Automatic construction of domain-specific sentiment lexicon based on constrained label propagation," *Knowledge-Based Systems*, vol. 56, pp. 191-200, 2014.

[6] D. Tang, F. Wei, B. Qin, M. Zhou, and T. Liu, "Building large-scale twitter-specific sentiment lexicon: A representation learning approach," in *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers*, 2014, pp. 172-182.

[7] S. Tan and Q. Wu, "A random walk algorithm for automatic construction of domain-oriented sentiment lexicon," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12094-12100, 2011.

[8] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Sentiful: Generating a reliable lexicon for sentiment analysis," in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, 2009, pp. 1-6: IEEE.

[9] S. Wu, F. Wu, Y. Chang, C. Wu, and Y. Huang, "Automatic construction of target-specific sentiment lexicon," *Expert Systems with Applications*, vol. 116, pp. 285-298, 2019.

[10] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proceedings of the 2008 international conference on web search and data mining*, 2008, pp. 231-240.

[11] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai, "Automatic construction of a context-aware sentiment lexicon: an optimization approach," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 347-356.

[12] Y. Zhang, H. Zhang, M. Zhang, Y. Liu, and S. Ma, "Do users rate or review? Boost phrase-level sentiment labeling with review-level sentiment classification," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014, pp. 1027-1030.

[13] A. Fahrni and M. Klenner, "Old wine or warm beer: Target-specific sentiment analysis of adjectives," in: Proceedings of the symposium on affective language in human and machine, AISB, 2008. p. 60–63.

[14] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267-307, 2011.

[15] P. D. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Transactions on Information Systems (TOIS)*, vol. 21, no. 4, pp. 315-346, 2003.

[16] S. Broumi, F. Smarandache, and M. Dhar, *Rough neutrosophic sets*. Infinite Study, 2014.

[17] S. Rizvi, H. J. Naqvi, and D. Nadeem, "Rough Intuitionistic Fuzzy Sets," in *JCIS*, 2002, pp. 101-104.

[18] S. Lee and G. Vachtsevanos, "An application of rough set theory to defect detection of automotive glass," *Mathematics and computers in simulation*, vol. 60, no. 3-5, pp. 225-231, 2002.

[19] F. Questier, I. Arnaut-Rollier, B. Walczak, and D. Massart, "Application of rough set theory to feature selection for unsupervised clustering," *Chemometrics and Intelligent Laboratory Systems*, vol. 63, no. 2, pp. 155-167, 2002.

[20] S. Hongchun, S. Xiangfei, and Y. Jilai, "A SURVEY ON THE APPLICATION OF ROUGH SET THEORY IN POWER SYSTEMS [J]," *Automation of Electric Power Systems*, vol. 3, 2004.

[21] J. Dai and Q. Xu, "Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification," *Applied Soft Computing*, vol. 13, no. 1, pp. 211-221, 2013.

[22] T. Slimani, "Application of rough set theory in data mining," *arXiv preprint arXiv:1311.4121*, 2013.

[23] M. Liu, M. Shao, W. Zhang, and C. Wu, "Reduction method for concept lattices based on rough set theory and its application," *Computers & Mathematics with Applications*, vol. 53, no. 9, pp. 1390-1410, 2007.

[24] H. Yu, G. Yang, M. Lin, F. Meng, and Q. Wu, "Application of rough set theory for NVNA phase reference uncertainty analysis in hybrid information system," *Computers & Electrical Engineering*, vol. 69, pp. 893-906, 2018.

[25] B. Tripathy, R. Mohanty, and T. Sooraj, "Application of uncertainty models in bioinformatics," in *Biotechnology: Concepts, Methodologies, Tools, and Applications*: IGI Global, 2019, pp. 141-155.

[26] R. Abooraig, S. Al-Zu'bi, T. Kanan, B. Hawashin, M. Al Ayoub, and I. Hmeidi, "Automatic categorization of Arabic articles based on their political orientation," *Digital Investigation*, vol. 25, pp. 24-41, 2018.

[27] D. H. Abd, A. T. Sadiq, and A. R. Abbas, "Political Articles Categorization Based on Different Naïve Bayes Models," in *International Conference on Applied Computing to Support Industry: Innovation and Technology*, 2019, pp. 286-301: Springer.

[28] D. H. Abd, A. T. Sadiq, and A. R. Abbas, "Classifying Political Arabic Articles Using Support Vector Machine with Different Feature Extraction," in *International Conference on Applied Computing to Support Industry: Innovation and Technology*, 2019, pp. 79-94: Springer.

[29] I. S. Al-Mejibli, D. H. Abd, J. K. Alwan, and A. J. Rabash, "Performance evaluation of kernels in support vector machine," in *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 2018, pp. 96-101: IEEE.

[30] I. S. Al-Mejibli, J. K. Alwan, and H. Abd Dhafar, "The effect of gamma value on support vector machine performance with different kernels," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, p. 5497, 2020.

[31] D. Zhang, "Bayesian Classification," in *Fundamentals of Image Data Mining*: Springer, 2019, pp. 161-178.

[32] S. Raschka, "Naive bayes and text classification i-introduction and theory," *arXiv preprint arXiv:1410.5329*, 2014.

[33] O. Oueslati, E. Cambria, M. B. HajHmida, and H. Ounelli, "A review of sentiment

analysis research in Arabic language," *Future Generation Computer Systems*, 2020.

[34] D. H. Abd, A. T. Sadiq, and A. R. Abbas, "PAAD: POLITICAL ARABIC ARTICLES DATASET FOR AUTOMATIC TEXT CATEGORIZATION," *Iraqi Journal for Computers and Informatics*, vol. 46, no. 1, pp. 1-10, 2020.

[35] M. Mustafa, A. S. Eldeen, S. Bani-Ahmad, and A. O. Elfaki, "A comparative survey on Arabic stemming: approaches and challenges," *Intel Inform Manage*, vol. 9, no. 2, pp. 39-67, 2017.

[36] M. Sawalha and E. Atwell, "Comparative evaluation of Arabic language morphological analysers and stemmers," in *Coling 2008: Companion volume: Posters*, 2008, pp. 107-110.

[37] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 275-282: ACM.

[38] Y. A. Al-Lahham, K. Matarneh, and M. Hasan, "Conditional arabic light stemmer: condlight," *Int. Arab J. Inf. Technol.*, vol. 15, no. 3A, pp. 559-564, 2018.

[39] K. Taghva, R. Elkhoury, and J. Coombs, "Arabic stemming without a root dictionary," in *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II*, 2005, vol. 1, pp. 152-157: IEEE.

[40] C. Sanderson and S. Guenter, "Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006, pp. 482-491: Association for Computational Linguistics.

[41] F. Peng, D. Schuurmans, and S. Wang, "Augmenting naive bayes classifiers with statistical language models," *Information Retrieval*, vol. 7, no. 3-4, pp. 317-345, 2004.

[42] A. Sharma, A. Nandan, and R. Ralhan, "An Investigation of Supervised Learning Methods for Authorship Attribution in Short Hinglish Texts using Char & Word N-grams," *arXiv preprint arXiv:1812.10281*, 2018.

[43] S. A. Taher, K. A. Akhter, and K. A. Hasan, "N-gram based sentiment mining for bangla text using support vector machine," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018, pp. 1-5: IEEE.

[44] N. Kumar and K. Srinathan, "Automatic keyphrase extraction from scientific documents using N-gram filtration technique," in *Proceedings of the eighth ACM symposium on Document engineering*, 2008, pp. 199-208: ACM.

[45] I. Hmeidi, M. Al-Ayyoub, N. A. Abdulla, A. A. Almodawar, R. Abooraig, and N. A. Mahyoub, "Automatic Arabic text categorization: A comprehensive comparative study," *Journal of Information Science*, vol. 41, no. 1, pp. 114-124, 2015.

[46] Q. A. Al-Radaideh and S. S. Al-Khateeb, "An associative rule-based classifier for Arabic medical text," *International Journal of Knowledge Engineering and Data Mining*, vol. 3, no. 3-4, pp. 255-273, 2015.

[47] R. Abooraig, S. Al-Zu'bi, T. Kanan, B. Hawashin, M. Al Ayoub, and I. Hmeidi, "Automatic categorization of Arabic articles based on their political orientation," *Digital Investigation*, vol. 25, pp. 24-41, 2018.

[48] Q. A. Al-Radaideh and M. A. Al-Abrat, "An Arabic text categorization approach using term weighting and multiple reducts," *Soft Computing*, vol. 23, no. 14, pp. 5849-5863, 2019.

[49] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 377-384.