Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Improving Sentiment Analysis of Arabic Tweets by One-way ANOVA

Manar Alassaf*, Ali Mustafa Qamar

Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia

ARTICLE INFO

Article history:
Received 15 June 2020
Revised 5 October 2020
Accepted 27 October 2020
Available online xxxx

Keywords:
Sentiment analysis
One-way ANOVA
Arabic tweets
Feature selection
Machine learning
High dimensionality

ABSTRACT

Social media is an indispensable necessity for modern life. As a result, it is full of people's opinions, emotions, ideas, and attitudes, whether positive or negative. This abundance of views creates many opportunities for applying sentiment analysis to the education sector, which reflects how countries and cultures develop. In this research, a real-world Twitter dataset was collected, containing approximately 8144 tweets related to Qassim University, Saudi Arabia. The main aim of this experimental study was to explore the possibility of using a one-way analysis of variance (ANOVA) as a feature selection method to considerably reduce the number of features when classifying opinions conveyed through Arabic tweets. The primary motivation for this research was that no previous studies had examined one-way ANOVA comprehensively to tackle the curse of dimensionality and to enhance classification performance in sentiment analysis for Arabic tweets. Therefore, various experiments were conducted to investigate the effects of one-way ANOVA and to select important features concerning the performance of different supervised machine learning classifiers. Support Vector Machine and Naïve Bayes achieved the best results with one-way ANOVA as compared to the baseline experimental results in the collected dataset. Furthermore, the differences between all results have been statistically analyzed in this study. As further evidence, one-way ANOVA with Support Vector Machine represented an excellent combination across different Arabic benchmark datasets, with its results outperforming other studies.

1. Introduction

In recent years, the global body of both structured and unstructured data has been growing exponentially—particularly because of the increasing popularity of social networking sites, which have generated massive amounts of data. This explosive data growth has made it challenging for organizations, institutions, governments, and individuals to acquire knowledge and make decisions to help them effectively benefit from the data. Twitter is one of the world's most popular microblogs, one which enables users to post tweets limited to 280 characters. Twitter is a global platform for expressing opinions, ideas, and feelings on various topics. It is worth exploiting whatever people have posted on it to develop marketing policies, support customer services, and improve various organizations and sectors.

Sentiment analysis (SA), used with textual data (Liu, 2012), is a technique for converting data into knowledge. SA includes a range of interrelated areas, such as Natural Language Processing (NLP), computational linguistics, and machine learning (ML). The primary purpose of SA is to extract subjective opinions from textual data. The classification of these opinions helps improve services based on the requirements of the beneficiaries, and this guides organizations and individuals to achieve the desired goals. SA can be divided into different levels of granularity: document level, sentence level, and aspect level (Liu, 2012). The chosen level depends on the purpose of the SA. The document level classifies the opinions into positive or negative sentiments in the document as a whole. On the other hand, sentence level separately classifies each sentence as positive, negative, or neutral. At this level, the sentence is the primary information unit. In research studies, tweets are usually treated as short texts similar to sentences (Kiritchenko et al., 2014; Paul and Borikar, 2018). The aspect level provide more comprehensive details than the previous levels. This level precisely determines what people like and dislike by analyzing the texts. In other words, it extracts the entities and their aspects to determine if the opinions concerning them are positive, negative, or neutral.

* Corresponding author.
E-mail addresses: a.manar@qu.edu.sa (M. Alassaf), al.khan@qu.edu.sa (A.M. Qamar).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2020.10.023
1319-1578/© 2020 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

Please cite this article as: M. Alassaf and A.M. Qamar, Improving Sentiment Analysis of Arabic Tweets by One-way ANOVA, Journal of King Saud University – Computer and Information Sciences https://doi.org/10.1016/j.jksuci.2020.10.023

The state-of-the-art approaches to SA are mainly categorized as supervised approaches (Hu et al. 2013) and lexicon-based (or unsupervised) ones (Assiri et al., 2018; Vu and Le, 2017). Lexicon-based approaches map the words either to a categorical (positive, negative, neutral) or a numerical score, which the algorithm uses to obtain the overall sentiment in the given text. On the other hand, supervised learning approaches depend on learning the classification model, which relies on a set of labeled data to teach the system how to generalize from the labeled training data to unseen situations. In the Arabic language, supervised approaches rather than their lexicon-based counterparts have been used in most of the SA studies (Alrefai et al., 2018). Some recent reviews consistent with the present authors' goal (Alrefai et al., 2018; Boudad et al., 2018; Mite-Baidal et al., 2018) have reported that Naïve Bayes (NB), k-Nearest Neighbor (k-NN), Support Vector Machines (SVMs), and Logistic Regression (LR) are the most popular ML classifiers for achieving SA tasks. The Multi-Layer Perceptron (MLP) has also been able to achieve excellent results in SA (Akhtar et al., 2017; Al-Batah et al., 2018).

By employing SA, the education sector can achieve significant improvements. With the opportunity to take advantage of big data, traditional forms for obtaining student or employee feedback are no longer very beneficial—especially since opinions are written spontaneously and explicitly in social media. Hence, studying a considerable amount of data published online is a matter of concern for the higher education domain. In the Arab world, there have been modest attempts to use SA in the education sector; for instance, Al-Rubaiee et al. (2016) implemented Arabic text classification to generate feedback from King Abdul-Aziz University students, using SVM and NB. They analyzed 1121 tweets, which were manually labeled into three classes (positive, negative, and neutral). The authors found that the best result was achieved by SVM in the positive and negative classes, but only while using the *n*-gram feature. To the best of our knowledge, their study, published in 2016, is the only research to have dealt with Arabic tweets in the education domain. However, the main weakness of their study was that the number of tweets they considered was the lowest when compared to similar studies (Chen et al., 2014; Abdelrazeq et al., 2015). Little attention has been devoted to employing SA on Arabic tweets in the education sector, and this omission, therefore, represented a strong incentive for this study's authors to collect tweets belonging to an educational entity.

Feature selection (FS) involves identifying a subset of important features for use in classification tasks. It significantly affects data mining in general and text mining in particular (Doan and Horiguchi, 2004). The main benefits of FS include facilitating data comprehension, decreasing the training time, and overcoming the curse of dimensionality. In Arabic text classification, some studies (Hawashin et al., 2013; Raho et al., 2015) have demonstrated that using the FS method greatly improves classification accuracy. There are four key FS methods, defined by their interactions with the learning model: filter method, wrapper method, embedded method, and the hybrid one. The filter method is a feature-ranking technique that evaluates the relevance of data features independent of the classification algorithm (Pervez and Farid, 2015). It counts on statistical methods and has low complexity. According to some previous studies (Elssied et al., 2014; Yang and Pedersen, 1997), the filter approach is widely used in the text classification field, where features are chosen by scoring matrices, such as information gain (IG), chi-squared, correlation coefficient, and analysis of variance (ANOVA). The wrapper method uses a specific algorithm to evaluate the quality of selected features and is a powerful way to tackle FS issues (Kohavi and John, 1997). A search procedure in the space of possible feature subsets is defined, and various feature subsets are generated and evaluated. Although the wrapper method is the most efficient among FS methods, its complexity and accuracy require more time (Ko et al., 2004). The embedded method integrates either a filter or a wrapper FS approach as well as a classifier into a single method for selecting important features. It has the advantages of (1) the wrapper method, which contains the interaction with the model, and (2) the filter method, which is less computationally intensive (Liu and Yu, 2005). Lastly, hybrid methods combine different approaches to get the best possible feature subset. One of the greatest advantages of hybrid methods is that they take the best advantages from other FS methods while reducing their disadvantages.

There have been four textual features that have been employed on Twitter data: semantic, syntactic, stylistic, and Twitter-specific features (Giachanou and Crestani, 2016).

- Semantic features: These are related to the meanings of words, such as sentiment words, opinion words, semantic concepts, and negation. Opinion, sentiment words, and phrases are the most used features in SA and can be extracted from lexicons.
- Syntactic features: These are unigrams, bigrams, *n*-grams, term frequencies, Part-of-Speech (POS), and dependency trees.
- Stylistic features: These have a non-standard writing style that is specific to social media. Some examples include emoticons and punctuation marks.
- Twitter-specific features: These are hashtags, retweets, replies, mentions, URLs, and tweet-length.

Most ML studies employ one-way ANOVA as a filter method to select the relevant features. One-way ANOVA has proven its effectiveness in solving the problem of high dimensionality in the feature space (Elssied et al., 2014; Grünauer and Vincze, 2015). A feature's variance determines its impact on the target class. If the variance is high, then there is a relationship between the feature and the target class. In other words, the target class is affected by features having a higher variance. To the best of our knowledge, no study has yet used one-way ANOVA comprehensively in SA. To date, only three studies (Arowolo et al., 2016; Elssied et al., 2014; Grünauer and Vincze, 2015) have used one-way ANOVA as an FS method, but none of them relates to the SA field. Therefore, this research used one-way ANOVA as an FS method in SA with the multi-class problem by reducing the high dimensionality of extracted features. The main contribution of this paper is to investigate the effectiveness of one-way ANOVA on SA at the sentence level. The proposed technique was applied to Arabic tweets related to Qassim University. The supervised approach was used with different ML classifiers, such as SVM, NB, LR, k-NN, and MLP.

The remainder of the paper is organized as follows. Section 2 introduces the background of ANOVA as an FS method and presents a literature review concerning the SA of Arabic tweets. The methodology is illustrated in Section 3. The experimental results are discussed in Section 4, and the conclusions and plans for future work are reported in Section 5.

2. Literature Review

There are two purposes of this section: first, to cover related works describing one-way ANOVA as a filter to reduce high dimensionality; and second, to discuss the state-of-the-art studies that have dealt with Arabic texts from the perspective of feature engineering.

ANOVA is a statistical method that decides whether the mean value of two or more groups is different (Stahle and Wold, 1989). It uses a probability distribution to measure the variance. In statistics, the probability value (*p*-value) is the probability of obtaining the observed results of a test. The *p*-value means the probability, assuming null hypothesis (H_0) is correct, that the test statistic equals the observed value or a value even more extreme in the direction predicted by the alternative hypothesis (H_1). The H_0 states that there is no difference among groups being studied, whereas H_1 states that there is a difference. The *p*-value is used as an option for rejecting H_0, according to the comparison of the results with the significance level. The significance level (α) is the probability of rejecting the null hypothesis when it is true. For example, an α of 0.05 implies a 5% danger of inferring that a difference exists while there is no actual difference. Thus, a smaller *p*-value implies that there is stronger evidence to support H_1 (Wasserstein and Lazar, 2016).

One-way ANOVA is a kind of ANOVA that is used as an FS filter to help measure the impact of a feature on a target class in ML tasks. As a univariate method, one-way ANOVA calculates a score for all features and then selects the features with the highest scores. One-way ANOVA has proven its effectiveness in solving the problem of high dimensionality in the feature space (Elssied et al., 2014; Grünauer and Vincze, 2015). A feature affects the target class if there is a difference between groups in terms of variance, which is the average of the squared differences from the mean. This leads to rejecting H_0, which states that all means of groups are equivalent, and accepting H_1. Deciding relevant features using ANOVA requires determining the threshold at which each feature is evaluated individually in terms of correlation with classes. Using one-way ANOVA as an FS filter helps to measure the impact of a feature on a target class. Consequently, each feature will have an *F*-value and a *p*-value as a score or weight. According to the calculated score, the important features will be determined. A higher *F*-value means a feature that impacts the class and will be considered relevant. Moreover, a low *p*-value, which is less than the significance level—0.05, as an example—will be recognized as an important feature. Some studies have used a percentage to choose the highest *F*-values of features, which are later forwarded to ML classifiers (Grünauer and Vincze, 2015). On the other hand, some studies have used the *p*-value to determine the important features of the target classes (Arowolo et al., 2016; Elssied et al., 2014).

In text classification, Elssied et al. (2014) proposed a novel spam detection approach by using a combination of FS based on one-way ANOVA and SVM to reduce the high dimensionality of textual data. Their experimental results demonstrated that the proposed method has high classification performance. One-way ANOVA as the FS method is fast, does not suffer from computational cost, and is simple to understand as well. Moreover, it is an approach that selects the features based on the difference between classes in terms of variance, regardless of the complexity of the language. Arabic was recently ranked as the fifth most often used language on the web, with approximately 133 million Arabic Internet users (Doochin, 2019). There are three types of Arabic: Classical Arabic (CA), Modern Standard Arabic (MSA), and informal Arabic (sometimes referred to as colloquial Arabic). CA is the language of Islam, which Arabic speakers use in their religious worship. MSA is the official language, which is understood by all Arabs, even if their religious affiliations are different. MSA is used in news bulletins, official speeches, and scientific articles. Informal Arabic is the language people use daily while speaking with their families and friends; this type of Arabic varies from region to region. Similarly, interaction through social media is widespread among Arab communities since it is an effective means of exchanging information on various topics and expressing opinions freely and frankly. Informal Arabic is usually used for writing on social media venues (Abdulla et al., 2014); however, many researchers in this field consider this to be an obstacle Alwakid et al., 2017). The analysis of the Arabic texts is extremely complicated, particularly with the diversity of dialects, where their use means ignoring the standard rules of grammar and spelling. (Albogamy and Ramsay, 2015) used POS taggers as syntactic features in Arabic tweets. The accuracy of standard Arabic taggers was 96–97% on MSA text, but their accuracy declined to 49–65% on Arabic tweets. Albogamy and Ramsay's work demonstrated that Arabic tweets do not follow the standard Arabic grammar. Consequently, dealing with Arabic tweets is a difficult task according to the NLP perspective. In SA, Al-Twairesh et al. (2018) used different methods of FS on the Saudi tweets benchmark dataset, called AraSenTi-Tweet created by Al-Twairesh et al. (2017). Those methods included three methods based on semantic features (all models' words, such as أحس/ المفروض/ يمكن; diminishers such as يعني/ أحيانا /تقريبا; and intensifiers such as جدا/ فعلا /خصوصا. In their study, the backward selection (backward elimination) algorithm was utilized to select the features. Backward selection starts with a complete set of features and then excludes features from the set, iteratively, until some stopping criterion is met. Hence, if the deletion of a feature set improves performance, this will lead to the deleted feature affecting the performance of classification negatively, and thus it should not be included in the final feature set. The main disadvantage of selection is its computational cost; however, it is acceptable for low-dimensional datasets. Al-Twairesh et al. tried using different task classifications: two-way classification, three-way classification, and four-way classification. Their experiment found that only two-way classification, which contained two sentimental classes (positive and negative), achieved the highest results with the specific sort of semantic features (all models' words). On the other hand, the best *F*-scores of three-way and four-way classifications were for stylistic features (emoticons): 60.71 and 53.56, respectively. Furthermore, one of the most popular methods in feature extraction that is not suffering from the problem of high dimensionality (only 100–300 dimensions irrespective of the number of words or samples) is Word2Vec. The Word2Vec method is a distributed representation learning algorithm used to learn continuous vector representations for words in an embedded low-dimensional vector space. It utilizes small neural networks to calculate word embeddings according to contexts of words. Vectors of words relocated in the vector space based on the words that have semantic similarities and share common contexts are mapped adjacent to each other in the space. Although different studies (Abu et al., 2019; Alali et al., 2019; Gridach et al., 2018) have used Word2Vec, which counts on the semantic similarity between words without any FS methods, there was a slight improvement in their results on selected Arabic benchmark datasets compared to other studies that were used in the comparison. The number of features, which ranged between 100 and 300, might not have been considered sufficient to obtain high performance in the text classification task.

3. Methodology

This section explains the framework methodology, which contains the major tasks performed to achieve the study's objectives.

SA was applied to Arabic tweets, and each tweet was classified into an appropriate opinion (positive, negative, or neutral). Fig. 1 shows the framework for applying SA to Arabic tweets at the sentence level.

According to Fig. 1, the main steps of implementation are dataset collection, pre-processing, feature extraction, FS, cross-validation, opinion classification, and performance measurements. These steps are described in the following subsections.

3.1. Dataset Collection

The first stage in dataset collection involved creating a Twitter Application Programming Interface (API) user to retrieve sufficient

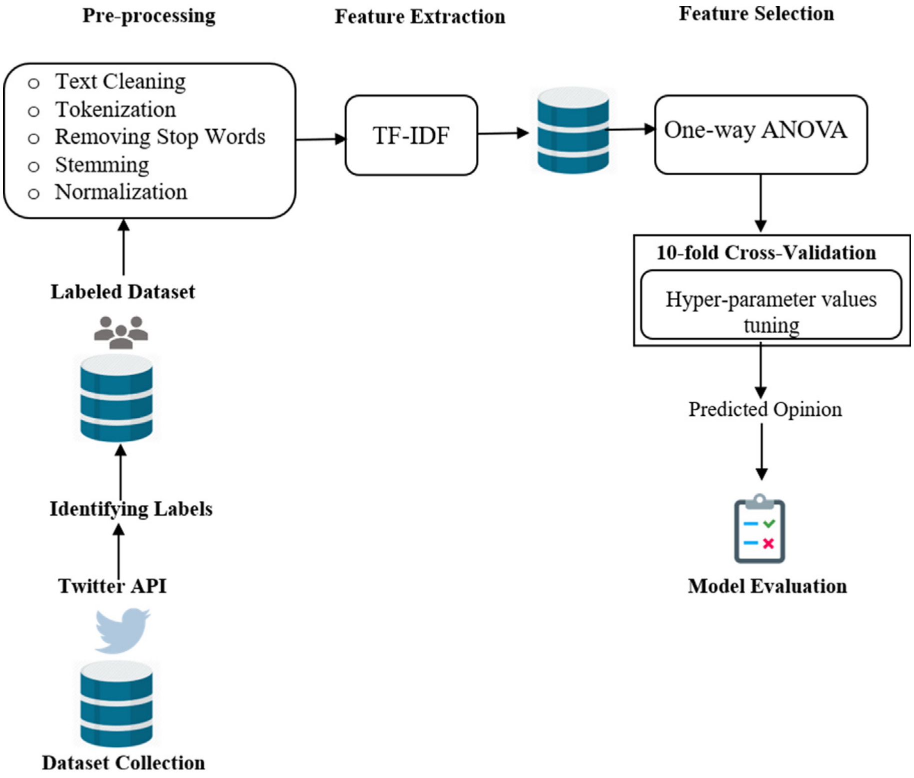

Fig. 1. The study's overall framework.

tweets. A keyword/hashtag combination was designed to collect the tweets. Thus, if any tweet contained جامعة القصيم or #جامعة_القصيم in its content, it was collected. Regarding the collection period, archived tweets were collected for more than 18 months, from January 2018 to June 2019. This period contained the Spring semester of the 2017–2018 academic year, and both semesters of the 2018–2019 academic year. A Python script collected the dataset by connecting with Twitter's official API to retrieve and save a list of tweets according to the mentioned conditions. The resulting dataset contained 67,659 tweets. Some filtering processes were implemented to extract the tweets that were most relevant to our study:

- Deleting duplicate copies of tweets in the dataset
- Deleting tweets containing advertisements that target university students
- Deleting tweets that only contained media, Uniform Resource Locators (URLs), or hashtags without any text

This reduced the number of tweets to 8234, on which the remaining steps in the methodology were implemented. Additionally, the authors noticed that most tweets were written in informal Arabic, especially in the Qassim dialect, although some tweets did appear in MSA.

Creating an annotated corpus is also essential for training ML algorithms. Thus, texts should be labeled to help machines understand them. In this study, the annotation task involved labeling each tweet according to its writer's opinion (positive, negative, or neutral). Three annotators conducted this process, all of whom were Qassim University graduates and native Arabic speakers. Whenever all annotators disagreed regarding a tweet's category, those tweets were excluded from the dataset before the training stage. Only 32 tweets were subject to this disagreement and were thereby excluded. The reason is that if they were confusing to all the three annotators, they would likely be misclassified by ML classifiers. After these 32 tweets were excluded, 8202 tweets remained. Table 1 shows a few examples of tweets after annotation.

3.2. Pre-processing

Pre-processing is the primary part of any text classification system since the words identified at this stage are the fundamental units passed on to subsequent processing stages. Pre-processing can improve data quality, thus enhancing the efficiency and accuracy of the mining process. The most popular pre-processing steps are text cleaning, tokenization, stop word removal, stemming, and normalization (Ghallab et al., 2020)). Most of the text written on social media is unstructured or noisy because of a lack of standardization or, the use of non-standard words, along with repetitions (Al-Shammari, 2009). When pre-processing the Arabic tweets for this research, the following tasks were performed:

1. Tweet cleaning: This process deletes unnecessary, insignificant items in texts (e.g., numbers, punctuation, URLs, special characters, non-Arabic letters, Twitter shortcuts, diacritics) to increase the classification performance.
2. Tokenization: This breaks up the text into individual words, i.e., tokens to identify the basic linguistic units for further processing.
3. Removing stop words: Stop words are the most common words in a language. For example, conjunctions, articles, and relational words are stop words in English. They are not useful in text classification because of their high frequency of occurrence. Therefore, their presence in text classification tasks presents an obstacle in understanding the textual data content and must be deleted. The Arabic language contains stop words that do not make real sense in the sentence, such as إذا, أنتم, عن, في.
4. Stemming: Stemming conflates different forms of a word into a single representation, referred to as the stem. For example, the words "writing," "wrote," and "writer," could all be reduced to a

Table 1
Examples of Inputs and Outputs of the Framework.

Inputs		Outputs
The Arabic Tweet	**Translation**	**Opinion**
وش هالاختبار الصعب والله حراااااام =(#جامعة_القصيم	That's not fair! The exam was so difficult =(#Qassim_University	Negative
يوم تغير المبنى الواحد ودّه يداوم كل يوم 😍♡ #جامعة_القصيم	Now, that the building has been changed, I want to come to the university everyday ♡😍 #Qassim_University	Positive
التقديم للدراسات العليا فتح في جامعة القصيم؟	Is the application for postgraduate studies in the Qassim University open yet?	Neutral

unique representation: "write." As mentioned earlier, most tweets in this dataset were written in the Arabic dialect. Stemming, however, is not accurate with dialectical words. It may result in either high or low classifier performance (Harrag et al., 2009). Arabic text mining performs better with a light stemmer than with a root stemmer (Sallam et al., 2016). The two most popular Arabic light stemmers are Tashaphyne and the one developed by the Information Science Research Institute (ISRI). With Saudi dialectal Arabic, previous research (Abozinadah and Jones, 2016) has achieved acceptable results with ISRI. Therefore, ISRI is used in this work.

5. Normalization: The main objective of normalization is to unify the shape of some Arabic letters that have different forms. It is similar to stemming but works at the level of a letter. Consequently, this phase significantly helps in the consistency of expected output. For example, it converts all forms of "alif" (إ, أ, آ) into ا, the different forms of "ya'a" and "alif maqsora" (ى,ي) into ي, the letter "ta'a" (ة) into ه, and the letter ؤ into و.

After the aforementioned pre-processing steps, the impact of this phase on tweets was similar to the example presented in Table 2.

As shown in Table 2, pre-processing made the tweets more consistent, without having noise or unwanted data. Thus, the tweets, now more predictable and analyzable, were ready to be passed on to the feature extraction phase.

After performing these steps, there was a chance of similar tweet content appearing in multiple tweets. Therefore, the tweets were checked to remove duplicates. Identical tweets do not contribute to learning a classifier; thus, extra copies should be deleted. After pre-processing, 58 duplicate tweets were removed, leaving 8144 tweets to be used in the experiments. Fig. 2 depicts the distribution of the opinion class in the dataset.

Neutral and negative classes make up the majority of tweets, at 4351 (53.4%) and 3128 (38.4%), respectively. The positive tweets have the lowest proportion, at merely 665 (8.2%).

Table 2
The Impact of Pre-processing on Tweets.

Before pre-processing	After pre-processing
هنيئاً لجامعة القصيم بتطبيق هذه الرسالة #توعية_وصحة_وتثقيف 👏👏 #جامعة القصيم	هني جمع قصم طبق رسل وعي وصح ثقف جمع قصم

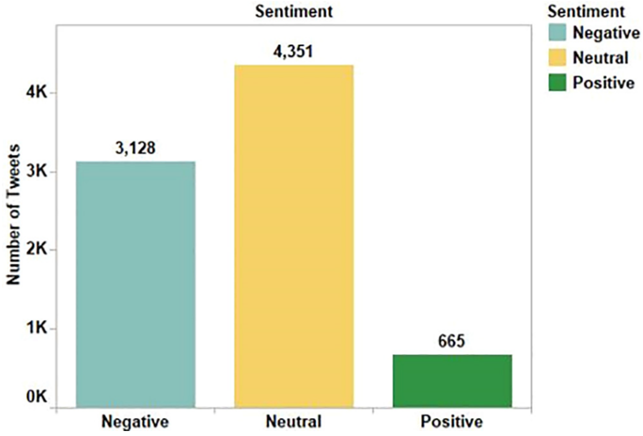

Fig. 2. Distribution of opinion classes in the dataset.

3.3. Feature Extraction

After the pre-processing stage, the textual content was ready to be converted into a numeric representation that could be understood by ML classifiers. This study uses the term frequency-inverse document frequency (TF-IDF). TF-IDF measures the term frequency—that is, the number of times a term occurs in a given document (Yamamoto and Church, 2001). This is particularly useful for text representation, where the word frequency indicates the important terms. In SA, the frequency of terms plays an essential role in identifying important information. Many words can frequently occur and could have a vital influence on identifying the polarity of the opinion. TF-IDF can be calculated by Eqs. (1)–(3).

$$w_{i,j} = tf_{i,j} \times idf_t \tag{1}$$

The first part of the equation contains term frequency (TF) that is the number of occurrences of the word in a given document The equation of TF is defined by Eq. (2).

$$tf_{i,j} = \frac{\text{Term } i \text{ frequency indocument } j}{\text{Total words indocument } j} \tag{2}$$

Furthermore, the inverse document frequency (IDF) is a part of the TF-IDF calculation, which provides higher weights for rare words and lower values for common words. The formula is given in Eq. (3):

$$idf_t = \log\left(\frac{\text{Total number of documents}}{\text{Documents with term } i}\right) \tag{3}$$

The FS method (a subsequent stage) is granted all possible options of feature forms (unigram, bigrams, trigrams) and the n-gram range was selected as [1–3]. n-gram means the number of sequence words in a given piece of text. According to this study (Giachanou and Crestani, 2016), the feature extraction method that was used extracted the syntactic features.

The dataset suffered from high dimensionality as the number of extracted features was 165,756.

3.4. Feature Selection

Two methods of one-way ANOVA, based on F-value and p-value, are used to statistically select the important features.

In the first one-way ANOVA method, the features were selected according to the F-values and based on the given percentile ($m\%$) of the original number of features. Only the $m\%$ top-scoring features were used to train the ML classifiers.

The second method depends on the p-values of one-way ANOVA, which determine the relevant features of the classification task and compare them to the significance level. If the p-value of a feature is less than the significance level, the feature is kept for further processing. Otherwise, it is discarded. The significance level (α) is usually set at 0.05 (Arowolo et al., 2016).

The methodology of selecting features for both one-way ANOVA methods is illustrated in Algorithms 1 and 2.

Algorithm1: Pseudocode of one-way ANOVA based on F-values.

```
INPUT: A pair $(E, Y)$, where $E$ represents features extracted by TF-IDF, and $Y$ is the classes of each feature. Also, percentage of the selected features $\%m$.
OUTPUT: A subset of features based on the F-value.
Begin
  $n\_classes \leftarrow$ Count $(Y)$
  For each $E_j \in (E, Y)$
    $n\_sample\_per\_classes \leftarrow$ Count $(Y_i)$
    $n\_samples \leftarrow$ Count $((E, Y))$
    $df_{bg} \leftarrow n\_classes - 1$ // Degrees of freedom between classes
    $df_{wg} \leftarrow n\_samples - 1$ //Degrees of freedom within classes
    $ss\_all\_features \leftarrow$ sum (square $(E)$)
    $sum\_all\_features \leftarrow$ sum $(E)$
    $square\_of\_sum\_all\_features \leftarrow$ square $(sum\_all\_features)$
    $SS_{total} \leftarrow ss_{all\_features} - \frac{square\_of\_sum\_all\_features}{n\_samples}$ //Total sum of squares
    $SS_{bg} \leftarrow 0$
    For each $Y_i \in Y$ do
      $SS_{bg} \leftarrow ss_{bg} + \frac{\text{square}(\text{sum}(E_{Y_i}))}{count((E,Y_i))}$
    End For
    $SS_{bg} \leftarrow ss_{bg} - \frac{square\_of\_sum\_all\_features}{n\_samples}$ //Sum of squares between classes
    $SS_{wg} \leftarrow SS_{total} - SS_{bg}$ //Sum of squares within classes
    $M_{sb} \leftarrow \frac{SS_{bg}}{df_{bg}}$ // Variance between classes
    $M_{wb} \leftarrow \frac{SS_{wg}}{df_{wg}}$ // Variance within classes
    $F - value \leftarrow \frac{M_{wg}}{M_{wg}}$ //Score of the feature
  End For
  Ascending Order $(E$based on $-value)$
  $FS \leftarrow$ Select (The highest $\%m$ of $E$ based on the $F - value$)
  Return $(FS)$
End
```

Algorithm 2: One-way ANOVA on p-values pseudocode

```
INPUT: A pair $(E, Y)$, where $E$ represents features extracted by TF-IDF, and $Y$ is the classes of each feature.
OUTPUT: A subset of features based on the p-value.
Begin
  $n\_classes \leftarrow$ Count $(Y)$
  For each $E_j \in (E, Y)$
    $n\_sample\_per\_classes \leftarrow$ Count $(Y_i)$
    $n\_samples \leftarrow$ Count $((E, Y))$
    $df_{bg} \leftarrow n\_classes - 1$ // Degrees of freedom between classes
    $df_{wg} \leftarrow n\_samples - 1$ //Degrees of freedom within classes
    $ss\_all\_features \leftarrow$ sum (square $(E)$)
    $sum\_all\_features \leftarrow$ sum $(E)$
    $square\_of\_sum\_all\_features \leftarrow$square $(sum\_all\_features)$
    $SS_{total} \leftarrow ss\_all\_features - \frac{square\_of\_sum\_all\_features}{n\_samples}$ //Total sum of squares
    $SS_{bg} \leftarrow 0$
    For each $Y_i \in Y$ do
      $SS_{bg} \leftarrow ss_{bg} + \frac{\text{square}(\text{sum}(E_{Y_i}))}{count((E,Y_i))}$
    End For
    $SS_{bg} \leftarrow ss_{bg} - \frac{square\_of\_sum\_all\_features}{n\_samples}$ //Sum of squares between classes
    $SS_{wg} \leftarrow SS_{total} - SS_{bg}$ //Sum of squares within classes
    $M_{sb} \leftarrow \frac{SS_{bg}}{df_{bg}}$ // Variance between classes
    $M_{wb} \leftarrow \frac{SS_{wg}}{df_{wg}}$ // Variance within classes
    $F - value \leftarrow \frac{M_{wg}}{M_{wg}}$
    $p - value \leftarrow \text{F\_survival}(\text{df}_{bg}, \text{df}_{wg}, F - value)$ //Score of the feature
    If $p - value < 0.05$ then
      Insert a feature into the set $FS$
  End For
  Return $(FS)$
End
```

One of the most remarkable differences between these methods is that the method based on F-scores requires determining the percentage of features, whereas the other method relies on a condition in selecting the features. Moreover, the p-value criterion is more stringent for filtering features than for the selected percentage method. Consequently, a set of features was chosen by the p-value, which may be a subset of the feature set determined by the percentage method.

3.5. Cross-validation Method

Five ML classifiers were implemented: SVM, NB, LR, k-NN, and MLP. These classifiers were employed with different experiments in order to discover the effectiveness of one-way ANOVA in classifier performance.

NB and LR are probabilistic algorithms that provide a probability distribution over output classes. On the other hand, the working of SVM is based on constructing the optimal hyperplane (decision surface) in the training phase to separate the data with the maximum generalization ability. The k-NN classifier is a case-based learning algorithm that uses a distance or similarity function for pairs of observations, such as the Euclidean distance or cosine similarity measures. Lastly, MLP is composed of perceptrons, stacked in various layers, to solve complex problems.

Some techniques deal with multi-class classification, such as one-against-rest (OAR) and one-against-one (OAO). The OAR approach considers one class as a positive one and the rest as negative in training the classifier. Therefore, for the data with n-classes, it trains n-classifiers. In the classification task, in order to classify an unseen example into a class, the highest probability from all the base classifiers is selected for the classification decision. In OAO, the approach considers each binary pair of classes and trains classifiers on a subset of data containing those classes. Thus, for the n-class, the approach generates the $n \times (n-1) \div 2$ binary classification problems. During the classification task, each classifier predicts one class only, and the final decision is given, for instance, by majority voting. Some ML algorithms can solve problems of multi-class classification by using OAR or OAO or can tackle this problem inherently, such as NB, k-NN, and MLP. The OAR approach is essentially a default technique for both SVM and LR, and it is accordingly used as such.

One of the most popular methods for tuning hyper-parameters is via cross-validation (CV), which is mentioned in the study by (Wainer and Cawley, 2018). In 10-fold CV, as the common CV but for each hyper-parameter setting, ten values of the performance measure are calculated. Then, the mean-tested performance measure is computed for each hyper-parameter setting. The highest average-tested performance measure serves as the final performance metric for the model. In this research, the hyper-parameters of each classifier were tuned according to the given values, as illustrated in Table 3.

The grid search algorithm is essentially an optimization algorithm used to select the values of hyper-parameters of a specific problem or algorithm that achieve the best results. The grid search algorithm is used in this study with a 10-fold CV to choose the best values of hyper-parameters for each classifier.

3.6. Model Evaluation

The F1-score is a popular metric in the multi-class classification problem. It is defined as the harmonic mean of precision and recall. Precision is the ratio of the correct predictions and the total predictions made by the ML classifier. Similarly, recall is the ratio of the correct predictions made by the ML classifier and the total number of correct SA classes. The macro average calculates the average across all classes, and it gives equal weight to all of the classes, no matter their size. The macro average of the F1-score is computed to evaluate the ML classifiers with features that were selected by one-way ANOVA.

4. Experimental Results

This section contains the results of the conducted experiments and the discussion of the findings, including the comparisons over multiple classifiers and scenarios. Furthermore, this section also includes the evaluation of the current study's results on different Arabic benchmark datasets to compare with other research studies that attempted to improve the performance of classifiers in the Arabic SA.

Table 3
Parameters of each classifier and their possible values in the tuning step.

ML classifier	Parameters long with the a range of values
SVM	C = [0.001, 0.01, 0.1, 1, 10, 100], Gamma = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100], Kernel = [sigmoid, linear, rbf]
NB	alpha = [0.00001, 0.0001, 0.001, 0.01, 0.1, 1], fit_prior = [True, False]
k-NN	n_neighbors = Range [1, 30]
LR	C = [0.001, 0.01, 0.1, 1, 10, 100], fit_intercept = [True, False]
MLP	learning_rate = [constant, adaptive], activation = [logistic, tanh, relu]

Table 4
Number of features used in each experiment.

#	Experiment	#Features
1	Baseline: Base	165,852
2	ANOVA_10%: One-way ANOVA (*F*-values) with 10% of the extracted features	16,585
3	ANOVA_20%: One-way ANOVA (*F*-values) with 20% of the extracted features	33,171
4	ANOVA_30%: One-way ANOVA (*F*-values) with 30% of the extracted features	49,755
5	ANOVA_*p*-value: One-way ANOVA (*p*-values)	19,594

4.1. Results and Findings

We considered five experimental scenarios, which could be distinguished based on the ranking of the features using one-way ANOVA. Firstly, the baseline experiment employed the methodology without the FS method. Then, one-way ANOVA, in the second, third, and fourth experiment, was used with 10%, 20%, and 30% of the original features, respectively, based on the highest *F*-values. In the fifth scenario, one-way ANOVA was used as an FS method based on the *p*-values of the features. Table 4 shows the impact FS had on the number of features during each experiment.

The number of features obtained from the last experiment accounted for 11.8% of the original features. The second experiment retrieved the least features—16,585 out of 165,852 features. In ANOVA 30%, 49,755 features were still a considerable number of features according to the number of samples, and that justified the reason why percentages of more than 30 were not considered.

The performance of the five classifiers (SVM, NB, LR, k-NN, and MLP) in all experiments is illustrated in Table 5.

As presented in Table 5, the evaluation results can be summarized as follows by comparing the results of the baseline experiments with those using one-way ANOVA as an FS method.

- SVM and NB outperformed all other classifiers, achieving exceptionally good results using one-way ANOVA as the FS method compared to their results in the baseline experiment. In the baseline experiment, both SVM and NB achieved an F1-score of 0.69.
- In the baseline experiment, MLP and k-NN achieved good results without using the FS method.
- The use of one-way ANOVA did not impact the results of LR compared to its results in the baseline experiment. The value of 0.70 for the F1-score is relatively constant.
- The experimental results of one-way ANOVA with *F*-values and *p*-values were quite close to each other at the classifier level for most of the classifiers.

4.2. Discussion

From a statistical perspective, one-way ANOVA and the Tukey test (Tukey, 1949) were used to determine whether there were significant differences between various results at the classifier and experimental levels. ANOVA was applied to 10-fold F1-score results. ANOVA is based on two assumptions: the normal distribution of the tested sample, and the homogeneity of variance in ANOVA, which requires that random variables have equal variance. Before applying ANOVA to examine the issue of significant differences, the assumptions associated with ANOVA were tested with these 10-fold F1-scores. The purpose of this statistical testing was to provide evidence that results are significantly different in terms of the *p*-value. If the *p*-value is smaller than the significance level, which is typically set at 0.05, then the difference between these results is statistically significant; otherwise, there is no statistically significant difference between the results. Therefore, if

Table 5
F1-scores of different experiments.

Classifier	Experiment				
	Base	ANOVA_10%	ANOVA_20%	ANOVA_30%	ANOVA_*p*-value
SVM	0.69 (±0.12)	**0.87 (±0.07)**	**0.86 (±0.08)**	**0.86 (±0.08)**	**0.88 (±0.07)**
NB	0.69 (±0.10)	0.78 (±0.09)	0.80 (±0.10)	0.82 (±0.09)	0.80 (±0.09)
LR	**0.70 (±0.13)**	0.70 (±0.12)	0.70 (±0.11)	0.70 (±0.11)	0.70 (±0.11)
k-NN	0.58 (±0.12)	0.44 (±0.08)	0.38 (±0.08)	0.36 (±0.08)	0.43 (±0.06)
MLP	0.67 (±0.12)	0.60 (±0.08)	0.61 (±0.10)	0.60 (±0.11)	0.60 (±0.10)

Table 6
Comparing the results with significant differences at the classifier level.

Classifier	**Experiments with Significant Differences**	***p*-value**
SVM	SVM_ ANOVA_10% vs. SVM_Base	0.000
	SVM_ANOVA_20% vs. SVM_Base	0.000
	SVM_ANOVA_30% vs. SVM_Base	0.000
	SVM_ANOVA_*p*-value vs. SVM_Base	0.000
NB	NB_ANOVA_10% vs. NB_Base	0.004
	NB_ANOVA_20% vs. NB_Base	0.000
	NB_ANOVA_30% vs. NB_Base	0.000
	NB_ANOVA_*p*-value vs. NB_Base	0.006
LR	No significant differences	–
k-NN	k-NN_ANOVA_10% vs. k-NN_Base	0.000
	k-NN_ANOVA_20% vs. k-NN_Base	0.000
	k-NN_ANOVA_30% vs. k-NN_Base	0.000
	k-NN_ANOVA_*p*-value vs. k-NN_Base	0.000
	k-NN_ANOVA_20% vs. k-NN_ANOVA_10%	0.018
	k-NN_ANOVA_30% vs. k-NN_ANOVA_10%	0.004
	k-NN_ANOVA_*p*-value vs. k-NN_ANOVA_30%	0.021
MLP	MLP_ANOVA_10% vs. MLP_Base	0.022
	MLP_ANOVA_*p*-value vs. MLP_Base	0.034

Table 7
The best F1-score results for each classifier.

Classifier	**The Best Scenario: F1-score**
SVM	All experiments with one-way ANOVA: 0.86–0.88
NB	All experiments with one-way ANOVA: 0.78–0.82
LR	All experiments with and without one-way ANOVA: 0.70
k-NN	Without one-way ANOVA: 0.58
MLP	Without one-way ANOVA: 0.67

using one-way ANOVA as an FS impacts the classification performance, there will be statistically significant differences between the results of the baseline experiment and those of other experiments. In addition to one-way ANOVA, the Tukey test is generally used to compare all classifiers with each other, determining which classifiers differ.

Table 6 presents the results of each classifier containing statistically significant differences based on the F1-score.

In SVM, statistically significant differences are observed between the baseline experiment and those using one-way ANOVA. The lowest SVM result was 0.69 (F1-score) in the baseline experiment. Hence, it may be said that the SVM classifier is an excellent option when ANOVA is used as the FS method. On the other hand, there were no significant differences between the remaining findings for SVM. The same conclusion could be drawn for the NB results. These findings reinforced the usefulness of ANOVA as the FS method with SVM and NB.

Additionally, the results of LR showed no improvement with ANOVA, and there were no significant differences between the baseline results and those of the ANOVA-based experiments. For the k-NN classifier, there were significant differences in all experiments with the FS method and baseline results. The best result for k-NN was found during the baseline experiment, leading to the conclusion that ANOVA did not improve the performance of k-NN. Lastly, for the experiments of MLP, the baseline result as compared to those of the ANOVA_10% and ANOVA_*p*-value experiments was statistically different.

Based on the aforementioned analysis, Table 7 shows the best experiments for each classifier individually.

SVM and NB, in the experiments with one-way ANOVA as the FS method, noticeably outperformed their findings during the base experiment. The best F1-score results for SVM were found in all experiments with ANOVA: 0.87, 0.86, 0.86, and 0.88, respectively. The results of these experiments were close, but the number of selected features was completely different, as mentioned in Section 4.1. The F1-score of SVM in the ANOVA_10%, ANOVA_20%, ANOVA_30%, and ANOVA_*p*-value experiments was enhanced by approximately 26.1%, 24.6%, 24.6%, and 27.5%, respectively, from its result in the base experiment (0.69). NB also performed very well in all one-way ANOVA experiments, achieving F1-scores of 0.78, 0.80, 082, and 0.80, respectively. The results with NB were enhanced by approximately 13.0%, 15.9%, 18.8%, and 15.9%, respectively, in the ANOVA experiments compared to its baseline F1-score.

The results with LR presented very similar F1-score values, reaching 0.70 with slight variations in the values of the standard deviation. Generally, the performance of LR in the experiments was constant and stable as compared to the changes that occurred in the results of other classifiers.

On the other hand, the results of k-NN and MLP were at their peak during the baseline experiments. The F1-score of k-NN and MLP was 0.58 and 0.67, respectively, with the baseline (without using one-way ANOVA). The MLP results of the remaining experiments were either 0.60 or 0.61 in terms of F1-score. Similarly, the F1-score results for k-NN ranged between 0.36 and 0.44 in all experiments of one-way ANOVA.

At the experiment level, Table 8 demonstrates the results without statistically significant differences, where the *p*-values were greater than or equal to 0.05.

In the scenarios for ANOVA_20% and ANOVA_*p*-value, there were significant differences between the F1-score results, which proved that using ANOVA for FS caused different impacts depending on the classifier. In ANOVA_10%, a comparison between LR (0.70) and NB (0.78) did not result in a significant difference. Additionally, statistical analysis showed no significant differences

Table 8
Results without significant differences at the experiment level.

Experiment	**Comparison**	***p*-value** > 0.05
Base	NB vs. SVM	1.000
	LR vs. SVM	0.999
	MLP vs. SVM	0.948
	LR vs. NB	0.999
	MLP vs. NB	0.948
	MLP vs. LR	0.850
ANOVA_10%	LR vs. NB	0.007
ANOVA_30%	NB vs. SVM	0.323

between the results of NB (0.82) and SVM (0.86) in the ANOVA_30% experiment. Finally, there were multiple comparisons between classifiers in the baseline experiment, such as NB (0.69) vs. SVM (0.69), LR (0.70) vs. NB (0.69), and MLP (0.67) vs. SVM (0.69). The single most striking observation emerging from the results of these comparisons was that SVM was the best classifier in all experiments using ANOVA. Except for the ANOVA_30% experiment, which provided 49,726 features, there were no significant differences between the results obtained with NB and SVM.

The classifiers differ in their working principles, and what may correspond to the work of one classifier may not be suitable for others. Therefore, the best experiment for achieving remarkable results will vary from one algorithm to another. Table 9 illustrates the best classifiers for each experiment.

The SVM results were among the best in all of the experiments. To the best of our knowledge, this is the first time one-way ANOVA has been used with both *F*-values and *p*-values to filter the features of Arabic tweets in the SA field. The findings confirm the usefulness of one-way ANOVA as an FS method in SA for Arabic tweets, especially with the SVM classifier, where the results after using ANOVA improved significantly. The use of one-way ANOVA with SVM provided very good results compared to other classifiers. NB also performed very well with ANOVA, and NB results showed noticeable improvement after using ANOVA as an FS method. The remaining classifiers showed no improvement in the results.

Table 9
The best classifiers for each experiment.

Experiment	The Best Classifier
Base	SVM (0.69), NB (0.69), LR (0.70)
ANOVA_10%	SVM (0.87)
ANOVA_20%	SVM (0.86)
ANOVA_20%	NB (0.82), SVM (0.86)
ANOVA_*p*-value	SVM (0.88)

4.3. External Comparison/Evaluation

In order to compare this study's results with other research works, three Arabic benchmark datasets were used, namely the Arabic Sentiment Tweets Dataset (ASTD), AraSenTi-Tweet, and Semantic Evaluation (SemEval-2017 Task 4). The detailed description of these datasets is provided in Table 10.

The experiments were carried out on the three mentioned datasets, and the results are shown in Table 11, which contains the F1-score for both macro and weighted averages. The weighted average computes the F1-score for each class and returns the average considering the number of samples for each class in the dataset. The weighted-average F1-score was computed because it is the main evaluation metric of ASTD; however, we computed it for all of the datasets. In ASTD, we used the same steps of pre-processing (only cleaning and tokenization) as Nabil et al.

According to Table 11, the best results of ASTD for different measures were achieved while using SVM in both ANOVA_20% and ANOVA_30% scenarios. The best results were 0.78 for the weighted-average F1-score in the case of ANOVA_30% and 0.58 for the F1-score (macro) in the same scenario. In addition to that,

Table 10
Descriptions of ASTD, AraSenTi-Tweet, and Semantic Evaluation (SemEval-2017 Task 4).

The Benchmark Dataset	#Tweets	#Classes	#Tweets for Training	#Tweets for Testing	Type of Arabic Language
ASTD (Nabil et al., 2015)	10,006	positive, negative, mixed, and neutral	6008 Training (positive:481, negative:1012, neutral:4015, mixed: 500), 1999 Validation (positive:159, negative:336, neutral:1338, mixed: 166),	1999 (positive:159, negative:336, neutral:1338, mixed: 166)	Egyptian dialect
AraSenTi-Tweet (Al-Twairesh et al., 2017)	15,751	positive, negative, and neutral	13,815 (positive:4235, negative:5515, neutral:4065)	1936 (positive:772, negative:640, neutral:574)	MSA and the Saudi dialect
SemEval-2017 (Task A) (Rosenthal et al., 2017)	9455	positive, negative, and neutral	3355 (positive:743, negative:1470, neutral:1145)	6100 (positive:1514, negative:2364, neutral:2222)	MSA and some Arabic dialects (Saudi, Egyptian, Syrian, etc.)

Table 11
Results of various classifiers while using one-way ANOVA on the three Arabic benchmark datasets.

Benchmark Dataset	Classifier	Experiment							
		ANOVA_10%		ANOVA_10%		ANOVA_10%		ANOVA_*p*-value	
		Macro	**Weighted**	**Macro**	**Weighted**	**Macro**	**Weighted**	**Macro**	**Weighted**
ASTD	SVM	0.57	0.72	**0.58**	0.75	**0.58**	**0.78**	0.55	0.74
	NB	0.42	0.67	0.43	0.68	0.42	0.67	0.43	0.68
	LR	0.47	0.65	0.47	0.68	0.46	0.68	0.48	0.67
	k-NN	0.23	0.55	0.22	0.55	0.21	0.54	0.24	0.56
	MLP	0.47	0.69	0.43	0.67	0.42	0.66	0.45	0.68
AraSenTi-Tweet	SVM	0.57	0.57	0.65	0.64	**0.70**	**0.69**	0.51	0.51
	NB	0.66	0.66	0.61	0.60	0.56	0.56	0.56	0.55
	LR	0.51	0.51	0.61	0.61	0.53	0.52	0.51	0.50
	k-NN	0.30	0.31	0.26	0.27	0.19	0.19	0.39	0.39
	MLP	0.62	0.62	0.64	0.64	0.60	0.60	0.62	0.62
SemEval-2017 (Task A)	SVM	0.56	0.56	0.68	0.66	**0.70**	**0.68**	0.51	0.50
	NB	0.49	0.51	0.41	0.45	0.42	0.46	0.48	0.50
	LR	0.49	0.50	0.50	0.50	0.49	0.49	0.51	0.51
	k-NN	0.41	0.43	0.23	0.26	0.21	0.24	0.36	0.37
	MLP	0.47	0.49	0.37	0.41	0.36	0.41	0.51	0.52

Table 12
Performance using one-way ANOVA compared to the state-of-the-art results across three benchmark datasets.

The benchmark dataset	Study	Types of features	F1-score (Macro)	F1-score (Weighted)
ASTD	(Nabil et al., 2015)	Syntactic	–	0.62
	(Al-Twairesh et al., 2018)	Semantic and backward selection algorithm	–	0.66
	(Gridach et al., 2018)	Semantic	–	0.72
	(Alali et al., 2019)	Semantic	–	0.76
	Present study (ANOVA_30%+SVM)	Syntactic and one-way ANOVA	0.58	**0.78**
AraSenTi-Tweet	(Al-Twairesh et al., 2017)	Syntactic	0.58	–
	(Al-Twairesh et al., 2018)	Stylistic and backward selection algorithm	0.60	–
	Present study (ANOVA_30%+SVM)	Syntactic and one-way ANOVA	**0.70**	0.69
SemEval-2017 (Task A)	(El-Beltagy et al., 2017)	Semantic and syntactic	0.61	–
	(Abu Farha and Magdy, 2019)	Semantic	0.63	–
	(Gridach et al., 2018)	Semantic	0.63	–
	Present study (ANOVA_30%+SVM)	Syntactic and one-way ANOVA	**0.70**	0.68

ANOVA_30% was the best scenario of the AraSenTi-Tweet dataset, where it achieved 0.70 for the macro-average F1-score and 0.69 for an F1-score (weighted). Regarding the results of SemEval-2017, the best result for both measures was in the ANOVA_30% scenario, accounting for 0.70 for the F1-score (macro) and 0.68 for the F1-score (weighted). It is worth mentioning that one-way ANOVA achieved the highest results with the SVM classifier compared to other classifiers, and this further strengthens our earlier conclusion that one-way ANOVA and SVM are a good combination. The comparison between the results obtained in this study and the results of state-of-the-art approaches is demonstrated in Table 12.

As shown in Table 12, our methodology which contained one-way ANOVA as the FS method, outperformed the current state-of-the-art models across all the datasets. In ASTD, the majority of studies have extracted semantic features from texts—except Nabil et al. (2015), which extracted syntactic features without using any FS method. Our approach undoubtedly outperformed other studies in the weighted-average F1-score term, achieving 0.78. Moreover, our findings in the AraSenTi-Tweet dataset were better than those in Al-Twairesh et al. (2017) and Al-Twairesh et al. (2018), which used syntactic, and stylistic features and the FS method, respectively. Although the best result was 0.60 by Al-Twairesh et al. (2018), we achieved 0.70. Furthermore, all studies that used SemEval-2017 (Task), extracted the semantic features; however, our approach was the best among all studies, where the result for F1score (macro) was 0.70.

The comparisons provide strong evidence for the effectiveness of using one-way ANOVA as an FS method, especially with SVM, to improve the performance and reduce the problem of high dimensionality in the SA of Arabic tweets.

5. Conclusion and Future Work

Sentiment Analysis (SA), as a text classification system, seeks to identify a writer's opinion through texts. Moreover, one-way ANOVA, as an FS method, has so far not been considered in the Arabic SA domain. Therefore, this study collected Arabic tweets written about a specific university and studied the feasibility of using one-way ANOVA in SA. The experiments in the FS stage were divided into two methods. The first one was based on ranking the features according to the specified percentage of the highest *F*-values in one-way ANOVA. Similarly, the second method worked by selecting the features based on *p*-values. Based on the results, it could be concluded that SVM and NB performed the best with one-way ANOVA, compared to the results obtained during the baseline experiment. On the other hand, the performance of LR was not affected at all while using one-way ANOVA. Lastly, the performance of k-NN and MLP degraded remarkably in trials containing one-way-ANOVA. The comparison between the results was supported with statistical evidence to guarantee that the findings did not occur by chance. Additionally, one-way ANOVA was also tested on three Arabic benchmark datasets, proving that ANOVA with SVM considerably improved the results of other studies. Generally, SVM and one-way ANOVA could be considered a great combination for obtaining the best SA results of Arabic tweets.

In the future, we plan to apply one-way ANOVA as the FS method along with the binary classification problem in SA. At a broader level, it can reduce the number of features substantially in SA by using either the wrapper or embedded method after implementing one -way ANOVA, as a hybrid FS method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdelrazeq, A., Janssen, D., Tummel, C., Jeschke, S., Richert, A., 2015. Sentiment Analysis of Social Media for Evaluating Universities, in: Proceedings of The Second International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC2015), Dubai, UAE. pp. 49–62.

Abdulla, N., Majdalawi, R., Mohammed, S., Al-Ayyoub, M., Al-Kabi, M., 2014. Automatic lexicon construction for Arabic sentiment analysis. IEEE, Barcelona, Spain, pp. 1–5. https://doi.org/10.1109/FiCloud.2014.95.

Abozinadah, E.A., Jones, J.H., 2016. Improved micro-blog classification for detecting abusive arabic twitter accounts. Int. J. Data Min. Knowl. Manag. Process (IJDKP) 6 (6), 17–28. https://doi.org/10.5121/ijdkp.2016.6602.

Abu Farha, I., Magdy, W., 2019. Mazajak: An Online Arabic Sentiment Analyser, in: Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy. pp. 192–198. https://doi.org/10.18653/v1/w19-4621.

Akhtar, S., Kumar, A., Ghosal, D., Ekbal, A., Bhattacharyya, P., 2017. A Multilayer Perceptron based Ensemble Technique for Fine-grained Financial Sentiment Analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark. pp. 540–546. http://dx.doi.org/10.18653/v1/D17-1057.

Alali, M., Sharef, N.M., Murad, M.A.A., Hamdan, H., Husin, N.A., 2019. Narrow convolutional neural network for arabic dialects polarity classification. IEEE Access 7, 96272–96283. https://doi.org/10.1109/ACCESS.2019.2929208.

Al-Batah, M.S., Mrayyen, S., Alzaqebah, M., 2018. Investigation of naive bayes combined with multilayer perceptron for arabic sentiment analysis and opinion mining. J. Comput. Sci. 14 (8), 1104–1114. https://doi.org/10.3844/jcssp.2018.1104.1114.

Albogamy, F., Ramsay, A., 2015. POS tagging for Arabic tweets, in: Proceedings of the International Conference Recent Advances in Natural Language Processing. Hissar, Bulgaria. pp. 1–8.

Alrefai, M., Faris, H., Aljarah, I., 2018. Sentiment analysis for Arabic language: A brief survey of approaches and techniques. arXiv Prepr. arXiv1809.02782.

Al-Rubaiee, H., Qiu, R., Alomar, K., Li, D., 2016. Sentiment analysis of arabic tweets in e-learning. J. Comput. Sci. 12 (11), 553–563. https://doi.org/10.3844/jcssp.2016.553.563.

Al-Shammari, E.T., 2009. A Novel Algorithm for Normalizing Noisy Arabic Text, in: Proceedings of 2009 WRI World Congress on Computer Science and Information Engineering, IEEE, Los Angeles, CA, USA. pp. 477–482. https://doi.org/10.1109/CSIE.2009.952.

Al-Twairesh, N., Al-Khalifa, H., Al-Salman, A., Al-Ohali, Y., 2017. AraSenTi-Tweet: A Corpus for Arabic Sentiment Analysis of Saudi Tweets, in: Proceedings of the 3rd International Conference on Arabic Computational Linguistics, ACLing, Dubai, United Arab Emirates, Procedia Computer Science, 117, Elsevier B.V. pp. 63–72. https://doi.org/10.1016/j.procs.2017.10.094.

Al-Twairesh, N., Al-Khalifa, H., Alsalman, A., Al-Ohali, Y., 2018. Sentiment Analysis of Arabic Tweets: Feature Engineering and A Hybrid Approach. arXiv Prepr. arXiv1805.08533.

Alwakid, G., Osman, T., Hughes-Roberts, T., 2017. Challenges in Sentiment Analysis for Arabic Social Networks, in: Proceedings of the 3rd International Conference on Arabic Computational Linguistics, ACLing, Dubai, United Arab Emirates, Procedia Comput. Sci., 117, Elsevier B.V. pp. 89–100. https://doi.org/10.1016/j.procs.2017.10.097.

Arowolo, M.O., Abdulsalam, S.O., Saheed, Y.K., Salawu, M.D., 2016. A feature selection based on one-way-anova for microarray data classification. Al-Hikmah J. Pure Appl. Sci. 3, 30–35.

Assiri, A., Emam, A., Al-Dossari, H., 2018. Towards enhancement of a lexicon-based approach for Saudi dialect sentiment analysis. J. Inf. Sci. 44 (2), 184–202. https://doi.org/10.1177/0165551516688143.

Boudad, N., Faizi, R., Thami, R.O.H., Chiheb, R., 2018. Sentiment analysis in Arabic: a review of the literature. Ain Shams Eng. J. 9 (4), 2479–2490. https://doi.org/10.1016/j.asej.2017.04.007.

Chen, X., Member, S., Vorvoreanu, M., Madhavan, K., 2014. Mining social media data for understanding students ' learning experiences. IEEE Trans. Learn. Technol. 7 (3), 246–259. https://doi.org/10.1109/TLT.2013.2296520.

Doan, S., Horiguchi, S., 2004. An efficient feature selection using multi-criteria in text categorization. IEEE, Kitakyushu, Japan, pp. 86–91. https://doi.org/10.1109/ICHIS.2004.20.

Doochin, D., 2019. How Many People Speak Arabic Around The World, And Where? [WWW Document]. Babble Mag. URL https://www.babbel.com/en/magazine/how-many-people-speak-arabic/ (accessed 10.1.20).

El-Beltagy, S.R., El Kalamawy, M., Soliman, A.B., 2017. NileTMRG at SemEval-2017 Task 4: Arabic Sentiment Analysis, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics (ACL), Vancouver, Canada. pp. 790–795. https://doi.org/10.18653/v1/s17-2133.

Elssied, N.O.F., Ibrahim, O., Osman, A.H., 2014. A novel feature selection based on one-way ANOVA F-test for e-mail spam classification. Res. J. Appl. Sci. Eng. Technol. 7 (3), 625–638 https://doi.org/10.19026/rjaset.7.299.

Ghallab, A., Mohsen, A., Ali, Y., 2020. Arabic Sentiment Analysis: A Systematic Literature Review. Appl. Comput. Intell. Soft Comput., 2020, Hindawi. pp. 1–21. https://doi.org/10.1155/2020/7403128.

Giachanou, A., Crestani, F., 2016. Like it or not: A survey of Twitter sentiment analysis methods. ACM Comput Surv 49, Article 28. pp. 1-41. https://doi.org/10.1145/2938640.

Gridach, M., Haddad, H., Mulki, H., 2018. Empirical evaluation of word representations on Arabic sentiment analysis, in: Proceedings of the International Conference on Arabic Language Processing: From Theory to Practice, ICALP, Communications in Computer and Information Science, vol 782. Springer, Cham. pp. 147–158. https://doi.org/10.1007/978-3-319-73500-9_11.

Grünauer, A., Vincze, M., 2015. Using Dimension Reduction to Improve the Classification of High-dimensional Data. arXiv Prepr. arXiv1505.06907.

Harrag, F., El-Qawasmeh, E., Pichappan, P., 2009. Improving Arabic Text Categorization using Decision Trees, in: Proceedings of the 2009 First International Conference on Networked Digital Technologies. IEEE, Ostrava, Czech Republic. pp. 110–115. https://doi.org/10.1109/NDT.2009.5272214.

Hawashin, B., Mansour, A.M., Aljawarneh, S., 2013. An efficient feature selection method for arabic text classification. Int. J. Comput. Appl. 83 (17), 1–6. https://doi.org/10.5120/14666-2588.

Hu, X., Tang, L., Tang, J., Liu, H., 2013. Exploiting Social Relations for Sentiment Analysis in Microblogging Categories and Subject Descriptors, in: Proceedings of the sixth ACM International conference on Web search and data mining (WSDM '13), Association for Computing Machinery, New York, NY, USA. pp. 537–546. https://doi.org/10.1145/2433396.2433465.

Kiritchenko, S., Zhu, X., Mohammad, S.M., 2014. Sentiment analysis of short informal texts. J. Artif. Intell. Res. 50, 723–762. https://doi.org/10.1613/jair.4272.

Ko, Y., Park, J., Seo, J., 2004. Improving text categorization using the importance of sentences. Inf. Process. Manage. 40 (1), 65–79. https://doi.org/10.1016/S0306-4573(02)00056-0.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artif. Intell. 97 (1–2), 273–324. https://doi.org/10.1016/s0004-3702(97)00043-x.

Liu, H., Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng. 17 (4), 491–502. https://doi.org/10.1109/TKDE.2005.66.

Liu, B., 2012. Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. Morgan & Claypool Publishers. https://doi.org/10.2200/S00416ED1V01Y201204HLT016.

Mite-Baidal, K., Delgado-Vera, C., Solís-Avilés, E., Espinoza, A.H., Ortiz-Zambrano, J., Varela-Tapia, E., 2018. Sentiment analysis in education domain: A systematic literature review, in: Proceedings of the International Conference on Technologies and Innovation. Springer, pp. 285–297. https://doi.org/10.1007/978-3-030-00940-3_21.

Nabil, M., Aly, M., Atiya, A.F., 2015. ASTD: Arabic sentiment tweets dataset, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Lisbon, Portugal. pp. 2515–2519. https://doi.org/10.18653/v1/d15-1299.

Paul, Y.R.S., Borikar, D.A., 2018. Sentiment analysis of tweets at sentence level using hadoop. Helix – Sci. Explor. 8 (5), 3797–3801.

Pervez, M.S., Farid, D.M., 2015. Literature review of feature selection for mining tasks. Int. J. Comput. Appl. 116 (21), 30–33. https://doi.org/10.5120/20462-2829.

Raho, G., Al-Shalabi, R., Kanaan, G., Nassar, A., 2015. Different classification algorithms based on arabic text classification: feature selection comparative study. Int. J. Adv. Comput. Sci Appl. 6 (2), 192–195. https://doi.org/10.14569/IJACSA.2015.060228.

Rosenthal, S., Farra, N., Nakov, P., 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, Canada. pp. 502–518. http://dx.doi.org/10.18653/v1/S17-2088.

Sallam, R.M., Mousa, H.M., Hussein, M., 2016. Improving Arabic Text Categorization using Normalization and Stemming Techniques. Int. J. Comput. Appl. 135 (2), 38–43. https://doi.org/10.5120/ijca2016908328.

Stahle, L., Wold, S., 1989. Analysis of variance (ANOVA). Chemom. Intell. Lab. Syst. 6 (4), 259–272. https://doi.org/10.1016/0169-7439(89)80095-4.

Tukey, J.W., 1949. Comparing individual means in the analysis of variance. Biometrics 5 (2), 99–114. https://doi.org/10.2307/3001913.

Vu, L., Le, T., 2017. A lexicon-based method for Sentiment Analysis using social network data, in: Proceedings of the International Conference on Information and Knowledge Engineering (IKE'17), Las Vegas, Nevada, USA. pp. 10–16.

Wainer, J., Cawley, G., 2018. Nested cross-validation when selecting classifiers is overzealous for most practical applications. arXiv Prepr. arXiv1809.09446.

Wasserstein, R.L., Lazar, N.A., 2016. The ASA statement on p-values: context, process, and purpose. Am. Stat. 70 (2), 129–133. https://doi.org/10.1080/00031305.2016.1154108.

Yamamoto, M., Church, K.W., 2001. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. Comput. Linguist. 27 (1), 1–30. https://doi.org/10.1162/089120101300346787.

Yang, Y., Pedersen, J.O., 1997. A Comparative Study on Feature Selection in Text Categorization, in: Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 412–420. https://dl.acm.org/doi/10.5555/645526.657137.