Natural Language Engineering (2024), **30**, pp. 1091–1129
doi:10.1017/S135132492300027X

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Cluster-based ensemble learning model for improving sentiment classification of Arabic documents

Rana Husni Al Mahmoud[1], Bassam H. Hammo[2,3] and Hossam Faris[2]

[1]Faculty of Information Technology, Applied Science Private University, Amman, Jordan, [2]King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan, and [3]King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan
Corresponding author: Bassam H. Hammo; Email: b.hammo@ju.edu.jo

(Received 5 November 2021; revised 8 April 2023; accepted 22 April 2023; first published online 1 June 2023)

Abstract

This article reports on designing and implementing a multiclass sentiment classification approach to handle the imbalanced class distribution of Arabic documents. The proposed approach, sentiment classification of Arabic documents (SCArD), combines the advantages of a clustering-based undersampling (CBUS) method and an ensemble learning model to aid machine learning (ML) classifiers in building accurate models against highly imbalanced datasets. The CBUS method applies two standard clustering algorithms: *K*-means and expectation–maximization, to balance the ratio between the major and the minor classes by decreasing the number of the major class instances and maintaining the number of the minor class instances at the cluster level. The merits of the proposed approach are that it does not remove the majority class instances from the dataset nor injects the dataset with artificial minority class instances. The resulting balanced datasets are used to train two ML classifiers, random forest and updateable Naïve Bayes, to develop prediction data models. The best prediction data models are selected based on F1-score rates. We applied two techniques to test SCArD and generate new predictions from the imbalanced test dataset. The first technique uses the best prediction data models. The second technique uses the majority voting ensemble learning model, which combines the best prediction data models to generate the final predictions. The experimental results showed that SCArD is promising and outperformed the other comparative classification models based on the F1-score rates.

Keywords: Arabic sentiment classification; Imbalanced dataset classification; Clustering-based undersampling; Ensemble learning model; SMOTE oversampling technique; Cost-sensitive classifier; Arabic-BERT language model

1. Introduction

Sentiment analysis or opinion mining is the task of predicting the sentiment polarity of a text (Pang, Lee, and Vaithyanathan, 2002). Recently, plenty of sentiment analysis research has already been conducted across different domains and at different levels (sentences, paragraphs, and documents) to motivate many natural language processing (NLP) applications and tools (Sadegh, Ibrahim, and Othman, 2012; Vinodhini and Chandrasekaran, 2012; Medhat, Hassan, and Korashy, 2014; Araque et al., 2017; Shayaa et al., 2018; Tedmori and Awajan, 2019). Predicting an opinion word is the main challenge of sentiment analysis. Once, it might be considered positive, while it might be negative in another context. A second challenge is that people express their opinions in different ways. However, most NLP applications are based on the fact that a bit of difference between two parts of a text might not significantly impact the meaning (Vinodhini and Chandrasekaran, 2012; Hussein, 2018). Generally speaking, there are two main approaches

used to build a sentiment analysis system: machine learning (ML) and lexicon-based (Biltawi et al. 2016; Alrefai, Faris, and Aljarah, 2018; Alayba et al. 2018; Mukhtar, Khan, and Chiragh, 2018; Verma and Thakur, 2018). A hybrid solution, which integrates the two approaches, has also been attempted (Zhang et al., 2011; Alrefai et al., 2018).

In the ML approach, the sentiment classification problem mainly depends on the prime ML algorithm. It is primarily based on extracting linguistic items with syntactic features (Taboada et al. 2011). In the corpus-based approach, also known as the supervised approach, decision tree (DT), k-nearest neighbor, Naïve Bayes (NB), and support vector machine (SVM) are applied to labeled datasets split manually into training and testing. The training dataset is used for building the model, while the testing dataset is used to evaluate its performance. For instance, the accuracy of a classification model is estimated by measuring the different types of errors made by a classifier (Abdulla et al., 2013). The lexicon-based approach involves summating the sentiment orientation of each word or phrase in the document (Turney, 2002; Taboada et al., 2011).

A dataset with imbalanced class distributions is problematic to many real NLP applications (He and Garcia, 2008; Kumar and Sheshadri, 2012; Bekkar and Alitouche, 2013). When imbalanced datasets are applied to classification tasks, there is a significant degradation in the performance of the most well-known classification algorithms. ML algorithms assume that the class distribution is relatively balanced and all misclassification costs are equal (Sun et al. 2007). However, several approaches and solutions have been proposed to address this problem (He and Garcia, 2008; Ganganwar, 2012; Ramyachitra and Manikandan, 2014). The attempts included enhanced approaches or models to handle imbalanced datasets (Bekkar and Alitouche, 2013) or to find the proper evaluation metrics for model assessment (Weiss, 2004).

Arabic is one of the under-resourced languages suffering from lacking quality resources in many computational research areas. Therefore, the main purpose of this study is to complement the existing research on Arabic sentiment analysis in general and to provide a solution to Arabic datasets with imbalanced class distribution in particular. The two main objectives of this study can be summarized as follows:

1. Investigate the various information sources to understand the sentiment analysis of Arabic text clearly.
2. Propose a solution to the problem of multiclass sentiment classification for handling imbalanced datasets to motivate research in Arabic sentiment analysis to build an effective sentiment prediction system for Arabic documents.

To achieve the goals, this study incorporates five stages. The first three stages include data collection, selecting the essential features from Arabic texts, and data preprocessing, which consists of steps that take as input a plain text document and outputs a set of tokens fed into the classification algorithms. The fourth stage involves conducting experiments to assess the performance of the synthetic minority over-sampling technique (SMOTE) and the Cost-Sensitive classifier and how they might handle the imbalanced problem in multiclass sentiment classification. Also, in this stage, we propose our hybrid approach, which combines the CBUS method and a majority voting ensemble model to aid ML classifiers in building accurate models against highly imbalanced datasets. The merits of the proposed approach are that it does not remove the majority class instances from the data nor injects the dataset with artificial minority class instances. The CBUS method uses two standard clustering algorithms: *K*-means (KM) and the expectation–maximization (EM), to balance the ratio between the major and the minor classes by decreasing the number of the major class instances and maintaining the number of the minor class instances at the cluster level.

The resulting balanced datasets are used to train two ML classifiers, random forest (RF) and updateable Naïve Bayes (UNB), to develop prediction data models. The best prediction data models are selected based on the accuracy rates. We applied two techniques to test our approach and

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

generate new predictions from the imbalanced test dataset. The first technique uses the best prediction data models. The second technique uses the majority voting ensemble learning model, which combines the best prediction data models to generate the final predictions. Finally, in the fifth stage, we discuss the experiments and their results. The key contributions of this research can be summarized as follows:

1. We introduce a multiclass sentiment classification approach for handling imbalanced class distribution of Arabic documents. The approach combines the advantages of a CBUS method with an ensemble learning model for improving the sentiment classification of Arabic documents (SCArD).
2. We evaluate the approach with state-of-the-art classification algorithms applied to imbalanced Arabic datasets of documents for automatic sentiment detection.

The rest of the paper is organized as follows. In Section 2, we provide a brief background and the related work on sentiment analysis for Arabic and English languages. Section 3 discusses the proposed model in more detail. In Section 4, we describe the conducted experiments to evaluate the performance of the proposed approach and discuss the results. Finally, we conclude our work and provide suggestions for future work.

2. Preliminaries and background

2.1 Preliminaries

Numerous clustering techniques for sentiment analysis have been proposed in the literature (Oueslati et al., 2020). Mainly, they fit into two categories: unsupervised lexicon-based and supervised ML. Both approaches rely on the bag-of-words model. In the lexicon-based approach, the unigrams from the lexicon are assigned a polarity score. The overall score of the text is computed as the sum of the polarities of the unigrams (Kolchyna et al., 2015; Bonta and Janardhan, 2019). While in the supervised ML, the classifiers use unigrams or a combination of n-grams as features to train and test the developed models. In addition, there is a hybrid approach, which combines the two approaches (Biltawi et al., 2016; Alrefai et al., 2018; Alayba et al., 2018).

Before we review the related work on sentiment analysis, we briefly describe the main methods and techniques used in this study to design and implement the multi-class sentiment classification model to handle the problem of imbalanced class distribution of Arabic documents.

Feature selection. The vast advancement of technology in recent years yields exponential data growth concerning both dimensionality and volume. Data management and automatic knowledge discovery of big data require the continual development of data mining and ML algorithms. Accordingly, the high dimensionality of data is considered a significant challenge to the ML algorithms (Tang, Alelyani, and Liu, 2014). One of the main challenges is overfitting. It is usually due to a massive number of irrelevant features that affect the learning algorithms' performance. Feature selection is necessary to address the problem of managing big data by reducing the dimensionality of features. The main feature selection task is to select a minimal subset of the relevant features from the original ones based on selection criteria. This will increase the performance of the learning algorithms, such as higher accuracy for classification, decrease the computational cost, and enhance model understanding (Tang et al., 2014). In general, feature extraction for sentiment analysis tasks can be applied at different levels of a text, such as:

- Document-level: predicting the sentiment of the whole document.
- Sub-document level: predicting the sentiment within a document section.
- Sentence-level: predicting the sentiment of a single sentence.
- Sub-sentence level: predicting the sentiment within a sentence.
- Title-level: predicting the sentiment of a title.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Morphological analysis and stemming. Morphological analysis and root extraction are essential for many Arabic NLP applications such as question answering, information retrieval, text summarization, and constructing Arabic corpora (Hammo et al., 2004; Hammo, 2009; Hammo et al., 2016). In the literature, a plethora of work has tackled the problem of Arabic morphological analysis (Al-Sughaiyer and Al-Kharashi, 2004; Boudlal et al., 2010; Pasha et al., 2014). In the morphological analysis and NLP applications, stemming is the process of reducing inflected and derived words to their word stem. Generally speaking, there are two approaches for Arabic stemming; a root-based approach described in (Khoja and Garside, 1999) and a shallow stemming approach described in (Larkey, Ballesteros, and Connell, 2002).

Classification models. The RF (Breiman, 2001) and NB (Ridgeway et al., 1998) algorithms are widely used for text classification and sentiment analysis (Singh, Singh, and Singh, 2017; Amrit et al., 2017; Hartmann et al., 2019; Kadhim, 2019; Khanday et al., 2020; Charbuty and Abdulazeez, 2021). Usually, they achieve high performance regarding the accuracy and F1-score rates. The Naïve Bayes updateable method is applied to improve the classification accuracy further. It is an updateable version of NB, also known as Flexible Bayes or Flex Bayes algorithm, and works in the same manner as the NB classifier (Mir et al., 2016).

Handling imbalanced datasets. In this study, we applied two techniques to deal with imbalanced datasets; the Oversampling via SMOTE and the cost-sensitive classification technique.

- ***Oversampling via SMOTE***. SMOTE is an over-sampling approach where the minority class is over-sampled by generating synthetic samples instead of oversampling with replacement (Chawla et al., 2002). Its main principle is to create new minority-class examples by interpolation among many minority-class examples that occur together. SMOTE uses standard Euclidean distance to find the k samples closest in the distance for each minority sample (He et al., 2008). After that, new synthetic samples are created by performing certain operations like rotation and skew (Batista, Prati, and Monard, 2004). By interpolation rather than replication, the overfitting problem can be avoided in SMOTE and causes the decision boundaries for the minority class to spread into the majority class space (Batista et al., 2004).
- ***Cost-sensitive classification***. Cost-sensitive classification considers the associated cost of misclassified examples rather than balancing distributions. This is done by considering the representative proportions of class examples in the distribution applied in sampling methods (Elkan, 2001). The objective of the cost-sensitive classification is to build and generate a model with the lowest cost by considering the cost matrix during building the model (Sun et al., 2007; He and Garcia, 2008). Furthermore, the cost matrix is always domain-dependent, and the defined costs can be different based on the application (Fernández et al., 2018).

Ensemble learning. Ensemble classifiers improve predictive ML results using constituent algorithms. They can mitigate many challenges, such as class imbalance and concept drift, as in many real-time ML applications, the distribution of features and the labels tend to change over time. The key idea of ensemble learning is to take an ensemble of "weak" learners and aggregate their results into one "strong" learner. An ensemble is considered a supervised learning algorithm (Whitehead and Yaeger, 2010; Xia, Zong, and Li, 2011; Bayoudhi et al. 2015). It has been proven experimentally that ensembles typically generate better results when there is significant diversity among the combined algorithms (Xia et al., 2011; Bayoudhi et al., 2015).

Clustering. Clustering is one of the most popular data mining tasks extensively studied in the context of the text to organize large volumes of text documents. It has a wide range of applications, including classification, visualization, and organization of text documents (AlMahmoud, Hammo,

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

and Faris, 2020). Text document clustering is essential in data indexing, information retrieval, managing, and mining extensive text data on the Web and incorporating information systems (Jing, 2008). Document clustering aims to group similar documents that form consistent clusters while differentiating the others. However, it is not a straightforward task to decide whether two documents are identical or not, as it mainly depends on the application (Huang, 2008). Selecting an appropriate clustering algorithm and evaluation metrics depends on the clustering objects and applications. KM and EM are conventional algorithms commonly used for text clustering (Singh, Tiwari, and Garg, 2011; Janani and Vijayarani, 2019).

Undersampling. Sampling is a class of methods that alters the size of training datasets. Undersampling and oversampling change a training dataset by sampling a smaller set of the majority of data and repeating instances in the minority data, respectively (Drummond and Holte, 2003). Undersampling is a popular method to deal with the imbalanced data problem. It uses only a subset of the majority class, and therefore it is very efficient (Liu, Wu, and Zhou, 2008).

2.2 Related work

Sentiment analysis for the Arabic language. In the following, we present the related work on sentiment analysis for the Arabic language. Next, the related work for English and other languages will be discussed. Table 1 compares the studies presented in this section. The related studies were organized based on the approaches they applied. They include lexicon-based, ML-based, hybrid, cluster-based, and CBUS. The other taxonomy that has been used consists of the language of the dataset, the evaluation metrics used to validate the efficiency of the approach, whether the dataset was balanced or not, and the techniques applied to solve the imbalanced problem if it existed.

Khoo and Johnkhan (2018) presented a survey of sentiment lexical construction approaches in detail. They classified them into four main techniques: (1) manual construction, (2) bootstrapping from a set of seed words, (3) adopting a lexicon from another domain using transfer learning, and (4) ML or probabilistic learning based on human sentiment coding.

Farra et al. (2010) introduced two approaches for predicting the sentiments of Arabic sentences. The first one considered the grammatical structure of a sentence. The second lexicon-based approach considered words of known sentiment orientation and their frequencies. The authors used the sentiments of different sentences from the same document to determine the sentiment of the entire document. Additionally, they used a dataset of Arabic movie reviews to evaluate their approaches. Assiri, Emam, and Al-Dossari (2018) and Al-Moslmi et al. (2018) also proposed a lexicon-based approach to enhance sentiment analysis of the Arabic language.

Shoukry and Rafea (2012) proposed a sentence-level sentiment analysis for Arabic based on ML algorithms. They applied the feature vectors to the NB and SVM Classifiers and compared the performance of the two classifiers to pick the classifier with the highest accuracy. Bayoudhi et al. (2015) proposed a supervised classification approach of Arabic documents. His approach embraced a multi-type feature set including opinion, stylistic, domain-dependent, and morpho-lexical features with discourse markers. A comparative study was conducted among a few state-of-the-art and ensemble-based classifiers with various combinations of algorithms. Alayba et al. (2018) combined CNNs and LSTMs networks and investigated their benefits to process Arabic sentiment classification. Because of the complexity of the orthography and morphology of Arabic, they used different levels of sentiment analysis to explore the effectiveness of the process. Other machine-based approaches to enhance sentiment analysis for the Arabic language were presented by El-Affendi, Alrajhi, and Hussain (2021) and Elfaik et al. (2021).

El-Halees (2011) proposed a hybrid approach made of three phases, including (1) applying a lexicon-based approach to classify documents, (2) using the classified documents from the lexicon-based method as a training set and then using the maximum entropy method to classify other documents, and (3) using the classified documents from the previous two phases as a

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 1. Comparison of sentiment analysis approaches for Arabic and other languages

Research	Language	Approach	Imbalanced (Y/N)	Techniques if imbalanced	Evaluation metrics
Assiri et al. (2018)	Arabic	Lexicon-based	N	-	Accuracy, precision, recall, and F1-score
Al-Moslmi et al. (2018)	Arabic	Lexicon-based	N	-	Macro-F1
Shoukry and Rafea (2012)	Arabic	Machine-based	N	-	Accuracy, precision, recall, and F1-score
Bayoudhi et al. (2015)	Arabic	Machine-based	N	-	Macro-averaged and F1-score
Alayba et al. (2018)	Arabic	Machine-based	N	-	Accuracy
El-Affendi et al. (2021)	Arabic	Machine-based	N	-	Accuracy
Elfaik et al. (2021)	Arabic	Machine-based	N	-	Precision, recall, and F1-score
Al-Azani and El-Alfy (2017)	Arabic	Machine-based	Y	Word embedding & ensemble learning	Accuracy, precision, recall, and F1-score
El-Halees (2011)	Arabic	Hybrid	N	-	Precision, recall, and F1-score
Taha (2017)	Arabic	Hybrid	N	-	Accuracy, precision, and recall
SCArD	**Arabic**	**CBUS**	**Y**	**Undersample & ensemble learning**	**Accuracy, precision, recall, and F1-score**
Taboada et al. (2011)	English	Lexicon-based	N	-	Percent correct
Aung and Myo (2017)	English	Lexicon-based	N	-	Average polarity score
Li et al. (2018)	English	Machine-based	Y	propose oversample technique	Accuracy and AUC
Ahmad et al. (2018)	English	Machine-based	N	-	F1-score
Xu et al. (2019)	English	Machine-based	N	-	Precision, recall, and F1-score
George and Srividhya (2022)	English	Machine-based	Y	Oversampling	Accuracy, precision, recall, and F1-score
Imran et al. (2022)	English	Machine-based	Y	Oversampling	Accuracy and F1-score
Madabushi et al. (2020)	English	Machine-based	Y	Word embedding	Precision, recall, and F1-score
Shaikh et al. (2021)	English	Machine-based	Y	-	BLEU, METEOR, ROUGEL, Skip-Thought and Embedding-Average
Rupapara et al. (2021)	English	Hybrid	Y	Oversampling	Accuracy, precision, recall, and F1-score
Dhillon et al. (2003)	English	Cluster-based	N	-	Accuracy

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 1. Continued

Research	Language	Approach	Imbalanced (Y/N)	Techniques if imbalanced	Evaluation metrics
Kyriakopoulou and Kalamboukis (2006)	English	Cluster-based	N	-	AUC
Yong et al. (2009)	English	Cluster-based	N	-	Precision, recall, and F1-score
Roul et al. (2015)	English	Cluster-based	N	-	Precision, recall, and F1-score
Onan (2017)	English	Cluster-based	N	-	F1-score
Chang et al. (2021)	English	Cluster-based	Y	Oversampling	G-mean
Jiang et al. (2022)	English	Cluster-based	Y	-	Accuracy, Recall, F1-score, and AUC
Li et al. (2011)	English	CBUS	Y	Undersampling	G-mean
Mountassir et al. (2012)	English	CBUS	Y	Undersampling	G-mean
Kim et al. (2021)	Korean	Machine-based	Y	Oversampling	Accuracy, precision, recall, and F1-score

training dataset. Finally, they applied the K-NN algorithm to classify the rest of the documents. Taha (2017) also proposed a hybrid approach for Arabic tweets sentiment analysis. The approach has two phases. The first phase used two weighting algorithms to assign high weights to the most significant features of the Arabic tweets. They include information gain and Chi-squared, and they were applied during the preprocessing phase along with stop-word removal, tokenization, and stemming. The second phase employed a learning technique to classify Arabic tweets as positive or negative. Their proposed approach was used on a dataset collected from Arabic tweets and has achieved higher accuracy and precision than other classification techniques such as SVM, DT, and NN. To address the problem of imbalanced data Al-Azani and El-Alfy (2017) applied the over-sampling technique on the minority class by adding synthetic samples using the SMOTE technique for Short Arabic Text.

Sentiment analysis for English and other languages. Taboada et al. (2011) extended their proposed dictionary, semantic orientation-calculator, to give polarity and strength to an opinion word. They computed semantic orientation using a simple aggregate-and-average method, where the total score of all adjectives was divided by the total number of adjectives in the document. Aung and Myo (2017) proposed a lexicon-based approach to analyze students' textual feedback to predict the performance of teaching faculty. The method was based on a manually created lexicon containing sentiment words and intensifiers. The presented results showed the sentiments of students at different levels of granularity.

In Yang and Chen (2017), the ML methods in sentiment analysis were summarized, and the formulas of traditional methods (such as SVM, NB, and ME) were also provided. In addition, they presented the latest ANN (BPN and CNN) methods. Finally, they gave the practical techniques and the challenges of emotion analysis. Ahmad et al. (2018) proposed an optimized sentiment analysis framework. They used the SVM grid search technique and the 10-k fold cross-validation to classify text. The grid search technique changes the gamma and costs parametric values of SVM. These values continually keep changing until the highest accuracy rate for a given dataset is reached. Xu et al. (2019) proposed a sentiment analysis system based on bidirectional long

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

short-term memory (BiLSTM) and applied it to the comment sentiment analysis task. The authors suggested an enhanced word representation approach, incorporating sentiment information using the classical term frequency-inverse document frequency (TF-IDF) and creating weighted word vectors. The comment vectors were better represented when the weighted word vectors were passed to BiLSTM. A feedforward neural network classifier determined the sentiment trend of BiLSTM the comment. Under identical conditions, the suggested sentiment analysis approach was compared against the sentiment analysis methods RNN, CNN, LSTM, and NB. The proposed approach has greater accuracy, recall, and F1-score rates based on the experimental results.

Many techniques were proposed to address the problem of imbalanced data (Li et al. 2011; Satriaji and Kusumaningrum, 2018; Ghosh et al., 2019). In addition, Rupapara et al. (2021) proposed a regression vector voting classifier, which is an ensemble strategy for identifying toxic comments on social media platforms. The ensemble combined the logistic regression with the support vector classifier using soft voting rules. They applied SMOTE for data balancing and TF-IDF and BoW for feature extraction. Kim, Koo, and Kim (2021) used an oversampling technique for imbalanced data.

George and Srividhya (2022) applied SMOTE for data balancing and combined ensemble-based bagging with SVM to enhance the classification performance.

Imran et al. (2022) used text generation models, CatGAN and SentiGAN, to generate new samples for minority groups. Also, they used deep learning and ML models to investigate the influence of synthetic text generation on the sentiment classification task for the highly imbalanced dataset.

Madabushi, Kochkina, and Castelle (2020) proposed a strategy that used cost-sensitivity with BERT to allow for improved generalization. They proposed a simple measure of corpus similarity to decide whether their approach was likely effective. Also, they reported that while BERT could handle imbalanced classes without extra data enrichment, it did not generalize effectively when the training and testing data were sufficiently diverse.

Shaikh et al. (2021) proposed a method that used text sequence-generating methods to solve the data imbalance problem. They combined GPT-2 with the LSTM-based text creation model to create artificial data. In the study, they examined three severely imbalanced datasets from various fields. They concluded that while GPT-2 works far better at the paragraph or document level than LSTM, LSTM performed well at the sentence level while producing synthetic text.

Li et al. (2018) presented a sentiment classification model to overcome two main challenges in sentiment classification; domain-sensitive categorization and data imbalance. The authors proposed a sentiment lexicon generation method using a label propagation algorithm. They utilized the generated lexicon to obtain synthetic samples for the minority class by randomly replacing a set of words with words of similar semantic content.

In the text classification literature, clustering was used either as an alternative approach for term selection to reduce the dimensionality (Dhillon, Mallela, and Kumar, 2003) or as a technique to enhance the training dataset. In the second case, clustering was used to discover a structure in the training examples and to expand the feature vectors with new attributes extracted from the clusters (Kyriakopoulou and Kalamboukis, 2006; Yong, Youwen, and Shixiong, 2009; Onan, 2017). Kyriakopoulou and Kalamboukis (2006) proposed an algorithm to combine supervised and unsupervised classification. In the unsupervised case, the aim was to extract a structure from a sample of objects or rephrase it appropriately to learn a concise representation of these data. The training and testing examples were clustered before the classification process to extract the structure of the whole dataset. Roul, Gugnani, and Kalpeshbhai (2015) proposed a clustering-based feature selection technique for text classification. First, the traditional KM clustering algorithm was applied to each dataset to generate k sub-clusters. Next, the important features of each sub-cluster were extracted using WordNet and TF-IDF scores. Finally, the top features were combined to generate the final reduced feature vector. The feature vector was used to train the ELM and ML-ELM classifiers. Chang, Chen, and Lin (2021) proposed the modified cluster-based over-sampling method for imbalanced sentiment classification. Jiang et al. (2022) proposed the KSCB model, which

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

combined *K*-means++, SMOTE, CNN, and Bi-LSTM. The proposed model employed CNN-Bi-LSTM to extract local features of text sentiment and capture context dependencies in sentences. It employed *K*-means++-SMOTE (a combination of *K*-means++ and SMOTE) to cluster the text sentiment and reduce between-class and within-class imbalances. The Kmeans++-SMOTE operation in KSCB was used to cluster sentiment text and then to build new corpora using an imbalance ratio to adjust data distribution. End-to-end learning was constructed using the loss function between *K*-means++-SMOTE and CNN-Bi-LSTM (combining CNN and Bi-LSTM).

Mountassir, Benbrahim, and Berrada (2012) proposed three different methods to under-sample the majority class of documents. These methods include removing similar, removing farthest, and removing by clustering. Almas and Ahmad (2007) proposed sentiment analysis methodologies for Arabic, Urdu, and English languages using computational linguistics. They discussed a local grammar method for extracting specialized terms automatically. Their experiments used a financial news dataset to evaluate their approaches. Abbasi et al. (2008) proposed a genetic algorithm for multilanguage sentiment classification. Document statistics and features measuring aspects of the writing style were used with word vectors to enhance a baseline classifier applied to a dataset of film reviews. These measures used syntactic and stylistic features such as word-length distributions, vocabulary richness measures, special-character frequencies, and character and word-level lexical features. They concluded that an entropy-weighted genetic algorithm could perform better than the standard feature reduction approach.

Despite the intensity of recent studies on sentiment analysis for the under-resourced Arabic language, as shown in Table 1, the biggest challenge is the lack of publicly available balanced benchmark datasets to validate newly developed methods or implemented software sentiment analysis systems. Recent work shows that researchers collected and built sentiment datasets from various social media platforms and manually analyzed them to extract sentiments. Although these efforts ended up with numerous solutions to the problem of sentiment analysis of the Arabic language, most datasets were imbalanced. Significant imbalanced data can affect the quality of the classification algorithms. Hence, most studies could not provide good generalizations for poorly represented classes.

To fill this gap in the Arabic language, this research presents the SCArD algorithm to handle the problem of imbalanced data in the SCArD. To test the efficacy of the SCArD algorithm, we used two sentiment classification datasets; the Gulf crisis and the Morocco-2016. Both datasets were appropriately annotated and human-verified (AlMahmoud et al., 2020). In addition, we tested the performance of SCArD on two publicly available datasets: the large-scale Arabic book review (LABR) dataset and the hotel Arabic-reviews dataset (HARD). Finally, we used the LABR and HARD datasets to compare the performance of SCArD using two feature extraction schemes; TF-IDF and the Arabic-BERT pre-trained language model. To the best of our knowledge, this is the first work combining CBUS and ensemble learning to solve this problem. This research can be helpful in advancing the research efforts in the field of sentiment analysis.

3. Research methodology

In this study, we followed a methodology made of five stages. It incorporated: (1) data collection, (2) feature extraction, (3) data preprocessing, (4) handling of imbalanced class distribution and model development, and (5) evaluation. Figure 1 shows the flow diagram of this methodology, whereas the following subsections discuss each stage in more detail.

3.1 Data collection

We used four datasets to test the SCArD algorithm; the Gulf crisis, the Morocco-2016, and two publicly available datasets; LABR and HARD. Subsection 3.2 describes the four datasets in more detail.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

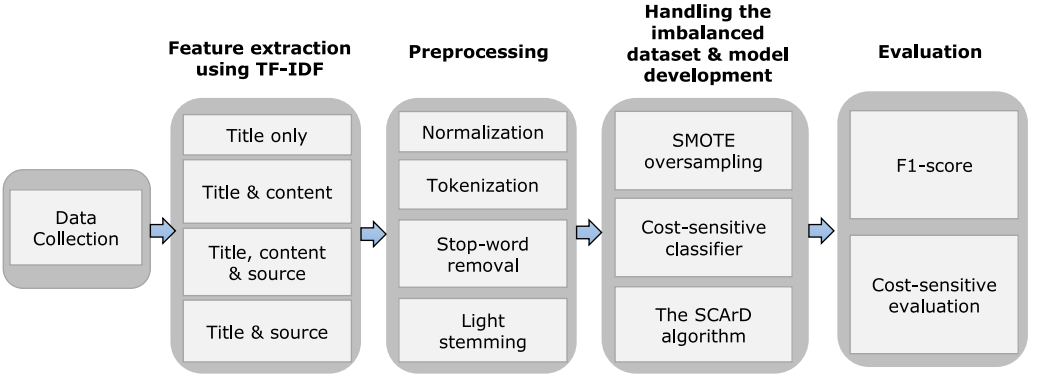

Figure 1. The flow diagram of the research methodology.

3.2 Datasets

The following is a description of the datasets we used to test the performance of the SCArD algorithm.

The Gulf crisis dataset. The dataset is about the Gulf crisis conflict that involved Qatar, United Arab Emirates, Saudi Arabia, and other regional countries. The dataset was collected and prepared by TooT,[a] which is a digital media company located in Amman, Jordan. The data collection was part of an ongoing research and investigation project on this conflict, and it is used in this research with their permission. A set of primary online news sources was selected to collect the related articles effectively based on their popularity. The sites were searched for relevant articles through two commonly used search engines: Google and Bing. Table 2 shows the online sources, which are websites of news agencies such as Reuters, news channels such as Aljazeera, or online versions of printed newspapers such as the Middle East. The number of the collected articles was (20,000). The articles that contain one of the search phrases in the title or the first two paragraphs were considered relevant and passed to the second phase for further processing and filtering. TooT's subject-matter experts categorized a set of (3161) relevant articles under one of the topics shown in Table 2. Next, the articles were manually labeled into negative, neutral, or positive sentiments. Two annotators annotated each article, and a supervisor monitored the annotation quality and discussed labeling disagreements as they arose. The inter-annotator agreement was 68%. This percentage is considered slightly reasonable because determining an article's sentiment is usually subjective to the judgment of the annotator. Detailed statistics of the dataset are shown in Table 3.

The Morocco-2016 dataset. The Morocco-2016 dataset is about the Morocco stereotype. The dataset is also the property of TooT. The data collection was part of ongoing research and investigation on the topic, and it was created for The Maghreb Center.[b] Further detail about the dataset can be found in (AlMahmoud et al., 2020). The dataset has (3520) articles, and they have been manually labeled into negative, neutral, and positive sentiments. Two subject-matter annotators from the Maghreb Center labeled each article, and a supervisor was available to mediate any labeling disagreements. The agreement between the annotators was 92%. Detailed statistics of the dataset are shown in Table 3.

[a]http://tootvs.com/
[b]http://maghrebcenter.org/

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 2. Sources and topics of news articles

Source name	URL	Source name	URL	Arabic Topic	Translation
الجزيرة	aljazeera.net	سكاي نيوز	skynewsarabia.co	السياسة	Politics
العربية	alarabiya.net	الشرق الاوسط	aawsat.com	الرياضة	Sports
سي ان ان العربية	arabic.cnn.com	صحيفة الحياة	alhayat.com	العلوم والتقنية	Sience and Technology
دوتش فيلية	dw.com/ar	سي ان بي سي	cnbcarabia.com	الاقتصاد	Economy
هافينغتون بوست	huffpostarabi.com	بي بي سي العربية	bbc.com/arabic	العلاقات الخارجية	Foreign affairs
فرنسا ٢٤	france24.com/ar	رويترز	ara.reuters.com	الثقافة والفن	Culture and arts
روسيا اليوم	arabic.rt.com	الخليج اون لاين	alkhaleejonline.net	السياحة	Tourism
عرب ٤٨	arab48.com	الخليج	alkhaleej.ae		

Table 3. Datasets

		Gulf crisis		Morocco-2016		LABR	
Sentiment class	Class label	Instances	pct	Instances	pct	Instances	pct
Negative	−1	350	11%	925	26%	8224	13%
Neutral	0	568	18%	556	16%	12,201	19%
Positive	1	2243	71%	2039	58%	42,832	68%
		3161		3520		63,257	

Table 4. The HARD dataset

		Original-HARD		Sampled-HARD	
Sentiment class	Class label	Instances	pct	Instances	pct
Negative	−1	52,849	13%	14,341	13%
Neutral	0	80,326	20%	21,798	20%
Positive	1	276,387	67%	75,000	67%
		409,562		111,139	

The LABR dataset. The LABR dataset was collected from the Goodread.com[c] website in March 2013 and is publicly available (Aly and Atiya, 2013). It contains over 63,000 book reviews in Arabic. The distribution of the imbalanced dataset of LABR is depicted in Table 3.

The HARD. The dataset was collected from the Booking.com[d] website during June/July 2016 and is publicly available (Elnagar et al., 2018). The reviews were expressed in modern standard and colloquial Arabic. The imbalanced dataset of HARD contains over 409,000 reviews, and its distribution is depicted in Table 4. Unfortunately, because the HARD dataset was so large that

[c]http://goodread.com/
[d]http://booking.com/

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

our computer machine could not handle it, we had to sample about 111,000 (27%) of the original dataset as given in Table 4.

Unlike the work of (Aly and Atiya, 2013) on the LABR dataset and the work of (Elnagar et al., 2018) on the HARD dataset where the researchers decided to neglect the neutral sentiments from all experiments, in this study, the neutral sentiments were found to be important and should not be neglected. The decision was also based on similar arguments discussed in (Koppel and Schler, 2006; Kaji and Kitsuregawa, 2007).

3.3 Feature extraction

3.3.1 The TF-IDF

Feature extraction is a process of knowledge discovery and dimensionality reduction. Features are extracted from documents according to their calculated weights in the collection. Many techniques can be used to extract features from a data collection (Gupta and Lehal, 2010). This study used two feature selection schemes; the statistical weighting scheme TF-IDF, and the Arabic-BERT pre-trained language model. TF-IDF determines the keywords that can identify or categorize some specific documents. It is defined as the product of TF(t, d) and IDF(t), where TF(t, d) is the number of times the word t occurs in document d, and IDF(t) is the inverse document frequency. It is calculated by Equation (1) (Jing, Huang, and Shi, 2002).

$$\mathrm{IDF}(t) = \log\left(\frac{N}{\mathrm{DF}(t)}\right) \tag{1}$$

Where N represents the total number of documents in the collection and the document frequency DF(t) is the number of documents in which the word t occurs at least once. The inverse document frequency is the highest if the word occurs only in one document, while its value is lowered if it occurs in too many documents. The value TF-IDF(t, d) of features t for document d is then calculated as a product value by Equation (2) (Jing et al., 2002).

$$\text{TF-IDF}(t, d) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t) \tag{2}$$

TF-IDF(t, d) is called the weight of word t in document d. This heuristic weighting scheme for a word says that if a word t frequently occurs in a document d (i.e., the term frequency is high), it is considered an effective indexing term for a document d. However, words that occur frequently but have low inverse document frequency are considered insignificant indexing terms (Jing et al., 2002).

3.3.2 The Arabic-BERT pretrained language model

BERT stands for Bidirectional Encoder Representations from Transformers. It is a neural network architecture introduced by Google in 2018 (Devlin et al., 2018). It is considered a state-of-the-art pretrained deep learning model for NLP tasks such as text classification, sentiment analysis, and question answering. BERT is an unsupervised learning model pretrained on large datasets such as Wikipedia and BookCorpus. It can be used directly, or it can be fine-tuned using a smaller labeled dataset to perform a specific supervised NLP task. Many transformers similar to BERT have been developed for different languages.

In this research, we used the Arabic-BERT-base model, which is an Arabic-BERT model pretrained on 8.2 billion words to learn contextualized representations of Arabic words and phrases (Safaya, Abdullatif, and Yuret, 2020). The data were collected from different resources, including the Arabic version of open super-large crawled aggregated coRpus,[e] Arabic Wikipedia,[f] and other

[e] https://traces1.inria.fr/oscar
[f] https://dumps.wikimedia.org/backup-index.html

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Arabic resources. The final dataset was about 95 GB of text. The development of Arabic BERT has greatly improved the ability of NLP systems to understand and generate Arabic text (Emami et al., 2022; Aftab and Malik, 2022).

The representation generated by Arabic-BERT is a vector that captures the meaning and context of the input text. This vector contains a lot of information about the words and their relationships to each other in the sentence. Algorithm 1 depicts how the vector representations for the input text were extracted from the output of the pretrained Arabic-BERT model.

Algorithm 1: *Vector representation extracted from the Arabic-BERT model*

```
Input: Processed documents(processedDocs)
Input: Arabic-base-BERT-model(ArBERTModel)
BEGIN
    For each document(d) in processedDocs
        Compute the vector representation:
        listOfWords[ ] = tokenize(d)
        sentenceVectorList[ ] = [ ]
        For each wordIndex(i) in listOfWords
            W^i=ArBERTModel.getVectorRepresentation(listOfWords[i])
            sentenceVectorList.append(W^i)
        EndFor
        sentenceVector=Mean(sentenceVectorList)
    EndFor
END
```

3.4 Data preprocessing

Data preprocessing includes a set of processes such as normalization of some Arabic letterforms, tokenization of words, removal of stop words, and stemming. In this phase, we applied the preprocessing steps described in a previous study of the first author (AlMahmoud et al., 2020) and the work of (Aref et al. 2020; Al-Laith and Shahbaz, 2021). All preprocessing tasks were handled automatically, and they include the following:

1. **Normalization** A set of normalization steps are usually applied to reduce the huge number of extracted terms. These steps include:
   - Removing non letters and special characters ($,&,%,. . .)
   - Removing non-Arabic letters
   - Replacing initial آ, إ or أ with bare alef ا
   - Replacing final ة with ه
   - Removing ال from the beginning of a word
   - Replacing final ي with ى
2. **Tokenization** Tokenization usually analyzes the text and splits it into a stream of individual tokens (words). It involves determining the boundaries of words, such as whitespaces and punctuation marks.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

3. **Removal of stop-words** Natural languages have their lists of stop words. For instance, in English, these words include articles such as "the, a, and an" and demonstratives like "this," "that" and "those," etc. Removing these high-frequency words from documents would decrease the number of indexed words and significantly improve the searching/retrieving time in many applications such as information retrieval. Likewise, the Arabic stop-words list includes words belonging to closed-class categories such as prepositions (عن,الى,. . .), demonstratives (هذا, هذه,. . .), adverbs (فوق, تحت,. . .), etc.
4. **Stemming** In this study, we adapted a shallow stemming approach, which removes the common affixes (i.e., prefixes and suffixes) from derivative words to extract their roots. It performs better than the root-based approach, which applies deep analysis to pull the roots.

3.5 The SCArD algorithm

The SCArD is given in Algorithm 2. It is mainly based on converting the imbalanced dataset into multiple balanced datasets and then training the classifiers separately on each of the new balanced datasets. This approach clusters the majority of class instances into several clusters using the CBUS method. It applied two commonly used clustering algorithms: KM and EM. Then it combines the instances of the minority classes with each cluster from the previous step. Each dataset should have a more balanced ratio of minority–majority classes. Finally, two classifiers (RF and UNB) are trained separately using the new balanced datasets to generate the best prediction data models based on accuracy rates. We applied two techniques to test our approach and generate new predictions from the imbalanced testing dataset. The first one uses the best prediction data models. The second uses the majority voting ensemble model, which combines the best prediction data models to generate the final predictions (Su et al., 2012; Rojarath, Songpan, and Pong-inwong, 2016). Figure 2 shows the workflow of the proposed approach, while the SCArD algorithm is given in Algorithm 2.

3.5.1 The clustered-based under-sampling

The merit of the CBUS technique is to balance the ratio between the majority and the minority classes of the imbalanced training dataset by decreasing the number of the majority class instances and maintaining the number of minority classes at the cluster level. As shown in Figure 2, the KM and the EM clustering algorithms were applied separately to the imbalanced training dataset of the majority class to cluster it into appropriate subsets of majority clusters to be merged with the instances on the minority classes. The number of clusters is determined experimentally based on the training dataset.

The workflow of SCArD, based on Figure 2 and Algorithm 2, can be summarized as follows:

1. The data split-phase. The imbalanced dataset (shown in Table 3) is split into two datasets; training (66%) and testing (34%). The two datasets are drawn using stratified sampling of the original dataset.
2. The initial training phase. The RF and the UNB classifiers are trained separately on the imbalanced training dataset. Both algorithms are trained using 10-fold cross-validation. The RF is trained 30 times, and the average of all evaluation metrics is taken, while the UNB is trained only once.
3. The training data split-phase. The imbalanced training dataset is divided into two datasets; the first one includes the majority class instances (positive class (1)), while the second dataset has all instances of the minority classes (neutral class (0) and negative class (−1)).

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

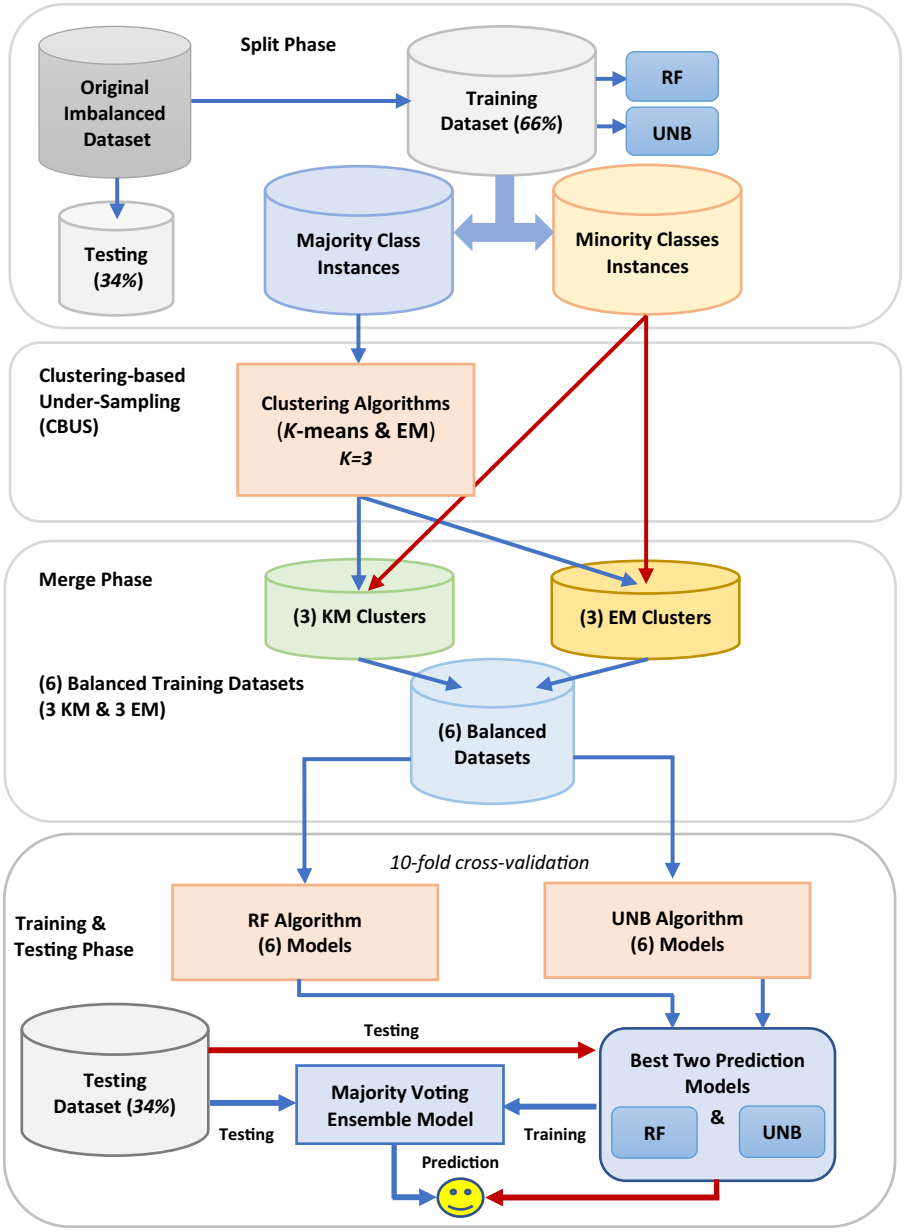

Figure 2. The workflow of the SCArD algorithm.

4. The CBUS phase. The CBUS method uses two common clustering algorithms: KM and EM. They are applied separately to the majority dataset. This phase developed six clusters; three clusters per each algorithm. The value $k = 3$ was determined experimentally.
5. The merge phase. The balanced datasets are generated after combining the dataset of the minority classes with each of the resulting clusters.
6. The training phase. The RF classifier is applied to each dataset using 10-fold cross-validation for 30 runs and taking the average of all evaluation metrics. Because the UNB is

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Algorithm 2: *The SCArD algorithm*

Input: Imbalanced dataset (D)

BEGIN

1. Split D into two datasets: Training (66%) and Testing datasets (34%).
2. Apply the RF and the UNB algorithms separately to the imbalanced training dataset and keep the two models for the majority voting ensemble step.
3. Split the Training dataset into majority (D_{maj}) and minority datasets (D_{min}).
4. Apply the CBUS method using the KM and the EM clustering algorithms separately to D_{maj}. This step developed six clusters; three per each algorithm. *K has been experimentally set to 3.*
5. Add D_{min} to each dataset produced from step 4. This step produces six balanced training datasets.
6. Apply the RF classifier to each balanced training dataset from step 5 using 10-fold cross-validation for 30 runs and take the average of all evaluation metrics. This step developed six prediction data models.
7. Repeat the previous step for the UNB classifier. This step also developed six prediction data models.
8. Select the best prediction data models of the RF and the UNB classifiers based on the accuracy rates. *The number of best models is determined experimentally.*.
9. Apply the best prediction data models from Step 8 to the imbalanced testing dataset to generate new predictions.
10. Apply the majority voting ensemble model, which combines the best prediction data models from step 8, and the RF and UNB models from step 2 to the imbalanced testing dataset to generate final predictions.

END

output: The best prediction results from steps 9 or 10

a deterministic algorithm, it takes only one run using 10-fold cross-validation. This phase developed 12 prediction data models, six models per classifier.

7. The testing phase. The best prediction data models (*determined experimentally*) from the training phase are selected based on accuracy rates. Next, they are applied to the imbalanced test dataset to generate new predictions. In addition, a majority voting ensemble model, which combines the best prediction models and the models from the initial training phase, is applied to generate the final predictions from the imbalanced test dataset.

3.5.2 SCArD complexity analysis

The overall complexity analysis of the SCArD algorithm is collectively based on running the classification algorithms shown in Table 5. The detailed phases and their complexities for running SCArD are given below.

1. Training–Testing splitting phase takes $O(n)$, where n is the number of documents.
2. The split phase takes $O(n)$, where n is the number of documents.
3. The clustering phase takes $O(knt) + O(k^2n)$ as shown in Table 5.
4. The merging phase takes $O(n)$, where n is the number of documents.
5. The classification phase takes $O(nlog(n)dr) + O(nd)$ as shown in Table 5.
6. The Ensemble phase takes $O(n)$, where n is the number of documents.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 5. Time Complexity of the SCArD algorithm

Algorithm	Time complexity
EM (Andrews and Fox, 2007)	$O(k^2n)^*$
K-means (Xu and Tian, 2015)	$O(knt)^*$
RF (Roy, Dey, and Chatterjee, 2020)	$O(nlog(n)dr)^*$
NB (Roy et al., 2020)	$O(nd)^*$

*d: number of features, k: number of clusters, n: number of documents, r: number of decision trees, t: number of iterations.

Table 6. The cost matrix of the Gulf Crisis problem

Class label	−1	0	1
−1	0	1	10
0	2.5	0	0
1	5	0	0

3.6 Evaluation metrics

To evaluate the performance of the developed classification models, we used two evaluation metrics; the F1-score and the cost-sensitive. The F1-score rate is the most common evaluation metric used in the literature for imbalanced datasets. In contrast, the cost-sensitive evaluation depends on a cost matrix. Because we are using an imbalanced dataset, the accuracy rate is not our concern for evaluating the proposed models.

The F-measure. For a class (C), the F1-measure value, also known as F1-score or simply F-score, is a composition of precision and recall. It is a consistent average of the two metrics which is used as an accumulated performance score. It is calculated by Eq. 3, which has been adapted from the general macro F1-score equation (Sokolova and Lapalme, 2009; Aref et al., 2020).

$$\text{F1-score (C)} = \frac{2 * \text{Precision(C)} * \text{Recall(C)}}{\text{Precision(C)} + \text{Recall(C)}} \tag{3}$$

Cost-sensitive measures. The objective of the cost-sensitive classification is to build and generate a model with the lowest cost by considering the cost matrix during building the model. The cost-sensitive evaluation value is calculated as the summation of all misclassified instances. Table 6 shows the most effective cost matrix depending on the evaluation metrics and the judgment of the subject-matter experts of Toot. The numbers −1, 0, and 1 represent the negative, neutral, and positive classes.

4. Experiments and results

In this section, we present the performance of the SCArD algorithm and compare it with other classification models. Also, we shed some light on the practical implications of this study on the future research of Arabic sentiment classification. First, we present the experimental setup. Next, the four main experiments conducted on the dataset are discussed. Finally, we discuss the evaluation performance of the SCArD algorithm.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

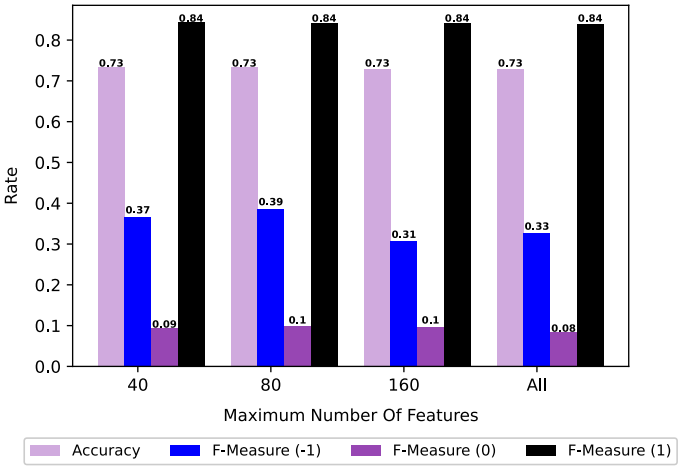

Figure 3. Performance of the RF classifier based on the number of extracted features.

4.1 Experimental setup

All experiments were conducted using a personal computer with Intel® core^{TM} i5-5500U CPU @ 2.53 GHz/4 GB RAM. The algorithms were implemented using the Java programing language. The WEKA library was used to run the classification algorithms and estimate the evaluation metrics.

4.2 Extraction of the best subset of features

Features (*tokens*) could be anywhere in a document. They were extracted experimentally from titles and content of documents after being preprocessed. The following discussion illustrates the approach we followed to extract the best features representing all documents in the dataset. For each document, we started with 40 features, 80, and 160. We ended up trying all document's features. Figure 3 shows the performance of the RF classifier using a different number of features. At 40 features, the F1-score rate for the negative class was (0.37) and for the neutral class was (0.09). At 80 features, the F1-score rate slightly improved to (0.39) and (0.1) for the same classes, respectively. However, there was a degradation in the F1-score rates when we selected a higher number of features. After all, we compiled four feature sets from the Gulf crisis dataset. Table 7 presents the characteristics of the feature sets. The first feature set FS_1 was extracted from the titles of the documents, FS_2 from titles and content, FS_3 from title, content, and source, and finally, FS_4 was extracted from title and source. The average number of features in the training dataset was 279. In this experiment, we did not report on the results of the UNB classifier because they were unsatisfactory.

4.3 Experiments

The experiments were conducted in four practical scenarios as follows:

- Experiment I: The effect of feature subset extraction on the classification process.
- Experiment II: Classification with SMOTE oversampling technique.
- Experiment III: Classification with a classifier combining cost-sensitive learning, RF, and UNB algorithms.
- Experiment IV: Classification with the SCArD algorithm.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 7. Feature sets characteristics

Feature set	Extracted from	Number of features (*tokens*)
Feature Set-1 (FS_1)	Title	830
Feature Set-2 (FS_2)	Title and content	1012
Feature Set-3 (FS_3)	Title, content, and source	1028
Feature Set-4 (FS_4)	Title and source	846

Table 8. The imbalanced training dataset after applying the SMOTE oversampling technique

	Negative class	Neutral class	Positive class
Original number of instances	245	398	1570
Number of instances after applying the SMOTE oversampling technique	1458	1568	
Percentage of oversampling	500%	300%	

- Experiment V: Classification with the SCArD algorithm: Comparing TF-IDF and Arabic-BERT pre-trained language model for feature extraction.

Experiments I, II, and III were conducted on the imbalanced Gulf crisis dataset, while experiment IV was conducted on four imbalanced datasets; the Gulf crisis, Morocco-2016, and the publicly available datasets: LABR and HARD. Experiment V was conducted only on the LABR and HARD datasets. In all experiments, the RF algorithm was trained using 10-fold cross-validation 30 times, then the average of all evaluation metrics was taken, while the UNB algorithm was trained only once. In the following subsections, we discuss the experiments in more detail.

4.3.1 Experiment I: The effect of feature subset extraction on the classification process

The first experiment was conducted on the imbalanced Gulf crisis training dataset. We ran the RF and UNB classifiers using the four feature sets described in Table 7 after fixing the number of features, experimentally, at 80. Table 9 shows the results of the two classifiers. The best results are in bold typeface. In this experiment, we observed that using the feature set (FS_4), which contains an article's title and source, provides the best F1-score rates for both classifiers and all classes (i.e., positive, neutral, and negative). For instance, the UNB classifier achieved an F1-score rate of (0.602) for the negative class, while the RF classifier achieved (0.867) for the positive class. The UNB classifier achieved (0.446) for the neutral class using the feature set (FS_3). This experiment indicates that the title gives a good sentiment about the article. This observation was investigated with the Gulf crisis dataset annotators and by taking samples from the documents. It was determined that Arabic titles usually use meaningful words, and unlike other languages, English, for example, Arabic titles rarely have abbreviations. Accordingly, combining the source of the text and the title enhanced the classification results.

4.3.2 Experiment II: Classification with SMOTE oversampling technique

The purpose of the second experiment is to study the effect of the SMOTE oversampling technique on the imbalanced training dataset. SMOTE was applied to the minor classes (negative class (−1)

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 9. The evaluation metrics of the RF and the UNB classifiers applied to the Gulf crisis feature sets

				Negative class			Neutral class			Positive class	
	Feature set	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
RF	FS1	0.713 ± 0.006	0.678 ± 0.033	0.261 ± 0.028	0.377 ± 0.032	0.360 ± 0.046	0.094 ± 0.014	0.149 ± 0.020	0.734 ± 0.004	0.959 ± 0.005	0.832 ± 0.004
	FS2	0.719 ± 0.004	0.711 ± 0.091	0.132 ± 0.029	0.222 ± 0.044	0.395 ± 0.087	0.038 ± 0.013	0.069 ± 0.023	0.725 ± 0.003	0.983 ± 0.003	0.834 ± 0.002
	FS3	0.733 ± 0.004	0.848 ± 0.050	0.209 ± 0.026	0.335 ± 0.036	0.472 ± 0.073	0.052 ± 0.013	0.094 ± 0.022	0.735 ± 0.003	0.988 ± 0.003	0.843 ± 0.003
	FS4	0.762 ± 0.006	0.789 ± 0.045	0.445 ± 0.039	0.567 ± 0.033	0.464 ± 0.046	0.140 ± 0.031	0.213 ± 0.038	0.779 ± 0.007	0.978 ± 0.005	**0.867 ± 0.004**
UNB	FS1	0.679	0.412	0.556	0.474	0.314	0.217	0.256	0.806	0.824	0.815
	FS2	0.686	0.411	0.631	0.498	0.376	0.461	0.414	0.875	0.752	0.808
	FS3	0.721	0.503	0.673	0.576	0.402	0.497	**0.445**	0.885	0.786	0.832
	FS4	0.736	0.539	0.681	**0.602**	0.378	0.358	0.367	0.870	0.844	0.857

The best results are in bold typeface.

and the neutral class (0)). Table 8 shows the oversampling ratios applied to the minor classes and the produced instances. Table 10 shows the effect of the SMOTE oversampling technique on the RF and UNB classifiers. The best results are in bold typeface. We observed that oversampling using SMOTE did not significantly improve the F1-score rates for both tested classifiers. However, the RF classifier showed a slight improvement in the F1-score rates for all classes using the feature set (FS_4). For instance, RF achieved an F1-score rate of (0.581) for the negative and (0.869) for the positive classes. The UNB classifier achieved (0.29) for the neutral class as shown in Table 10. The reported results were achieved at an oversampling ratio of 500% for the negative and 300% for the neutral classes.

4.3.3 Experiment III: Classification with a classifier combining cost-sensitive learning, RF, and UNB algorithms

The third experiment aims to study the effect of combining the cost-sensitive classifier with RF and UNB on the imbalanced training dataset. The results of this experiment using the cost matrix (shown in Table 6) are given in Table 11. The best results are in bold typeface. A closer look at Table 11 shows that combining the cost-sensitive classifier with the RF algorithm achieved the best F1-score rates when using the feature set (FS_4). For instance, RF achieved an F1-score rate of (0.624) for the negative and (0.868) for the positive classes. For the neutral class, when combined with UNB, it achieved an F1-score rate of (0.438) using the feature set (FS_3) as shown in Table 11. The results we obtained complied with the lowest calculated cost-sensitive values (Total Cost), shown in the third column of Table 11. Combining the cost-sensitive classifier with RF produced the lowest cost at (531.8) when using the feature set (FS_4), while for the UNB, the lowest cost was at (443) for (FS_3). This experiment reveals that using the cost-sensitive classifier combined with RF and UNB on the imbalanced training dataset slightly improved the F1-score rates of the negative class and moderately lowered them for the neutral class.

4.3.4 Experiment IV: Classification with the SCArD algorithm

Figure 2 describes the workflow of SCArD, while Algorithm 2 describes the training and testing processes. The CBUS method uses KM and EM clustering algorithms and applies them to the imbalanced training dataset. As discussed earlier, this step produces three clusters for each clustering algorithm. The number of clusters (K) was determined experimentally, and the best number was fixed at three. Nevertheless, we tried different values of K such as 5, 7, and higher; however, we always ended with either empty clusters or clusters with a few instances. The balanced datasets were generated after combining the dataset of the minor classes with each of the resulting six clusters.

Table 12 shows the number of instances of each cluster for both clustering algorithms (i.e., KM and EM) applied to the Gulf crisis dataset. In a similar way, the Morocco-2016 dataset is given in Table 13, the LABR dataset is given in Table 14, and the HARD dataset is given in Table 15.

As for the performance of the SCArD algorithm, Table 16 presents the RF, and the UNB classifiers applied to each balanced dataset resulting from the Gulf crisis dataset after using the CBUS method. As shown in Table 16, a few models showed better performance in terms of accuracy rates than the others. For example, models $RFKM_{C2}$, and $RFEM_{C1}$ were among the best models achieving the highest accuracy rates. Therefore, those two models can be used in the ensemble classification to improve the results further. After all, we applied the following steps to the imbalanced Gulf crisis test dataset:

1. The majority voting ensemble classifier combined the best two data models achieving the highest accuracy rates from Table 16 (i.e., $RFKM_{C2}$ and $RFEM_{C1}$) with the RF and UNB models, applied separately to the imbalanced training dataset, to build an ensemble model.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 10. The evaluation metrics of the RF and the UNB classifiers using SMOTE.

				Negative class			Neutral class			Positive class	
	Feature set	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
RF	FS1	0.706 ± 0.006	0.641 ± 0.034	0.303 ± 0.026	0.411 ± 0.025	0.313 ± 0.036	0.129 ± 0.021	0.183 ± 0.026	0.745 ± 0.004	0.931 ± 0.007	0.827 ± 0.004
	FS2	0.719 ± 0.004	0.670 ± 0.081	0.119 ± 0.019	0.202 ± 0.029	0.451 ± 0.111	0.040 ± 0.010	0.073 ± 0.018	0.724 ± 0.003	0.984 ± 0.003	0.834 ± 0.002
	FS3	0.727 ± 0.004	0.849 ± 0.074	0.150 ± 0.027	0.254 ± 0.040	0.487 ± 0.106	0.047 ± 0.013	0.085 ± 0.022	0.729 ± 0.003	0.990 ± 0.003	0.839 ± 0.002
	FS4	0.763 ± 0.005	0.755 ± 0.044	0.474 ± 0.036	**0.581 ± 0.026**	0.456 ± 0.027	0.192 ± 0.025	0.270 ± 0.026	0.793 ± 0.006	0.961 ± 0.006	**0.869 ± 0.004**
UNB	FS1	0.533	0.236	0.634	0.344	0.268	0.228	0.246	0.804	0.597	0.685
	FS2	0.72	0.558	0.233	0.328	0.413	0.070	0.12	0.740	0.964	0.837
	FS3	0.71	0.525	0.201	0.291	0.218	0.0409	0.068	0.738	0.962	0.835
	FS4	0.614	0.357	0.709	0.475	0.281	0.300	**0.290**	0.831	0.680	0.748

The best results are in bold typeface.

Table 11. The evaluation metrics of the RF and the UNB classifiers using the cost-sensitive classifier.

				Negative class			Neutral class			Positive class		
	Feature set	Accuracy	Total cost	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
RF	FS1	0.720 ± 0.005	827.067 ± 37.072	0.597 ± 0.031	0.355 ± 0.030	0.444 ± 0.028	0.344 ± 0.097	0.023 ± 0.009	0.044 ± 0.017	0.735 ± 0.005	0.970 ± 0.005	0.836 ± 0.003
	FS2	0.725 ± 0.005	842.250 ± 39.264	0.604 ± 0.061	0.249 ± 0.036	0.352 ± 0.042	0.401 ± 0.239	0.006 ± 0.003	0.012 ± 0.005	0.732 ± 0.004	0.981 ± 0.004	0.838 ± 0.003
	FS3	0.742 ± 0.004	716.300 ± 38.608	0.736 ± 0.040	0.354 ± 0.037	0.477 ± 0.037	0.303 ± 0.323	0.005 ± 0.005	0.010 ± 0.010	0.743 ± 0.004	0.990 ± 0.003	0.849 ± 0.003
	FS4	0.762 ± 0.005	**531.833 ± 37.046**	0.667 ± 0.033	0.589 ± 0.041	**0.624 ± 0.025**	0.505 ± 0.113	0.028 ± 0.007	0.053 ± 0.014	0.777 ± 0.007	0.984 ± 0.003	**0.868 ± 0.004**
UNB	FS1	0.685	829.5	0.374	0.686	0.484	0.384	0.114	0.176	0.804	0.837	0.820
	FS2	0.692	575	0.406	0.679	0.509	0.397	0.432	0.414	0.870	0.761	0.812
	FS3	0.724	**443**	0.470	0.692	0.560	0.421	0.456	**0.438**	0.881	0.798	0.837
	FS4	0.729	490	0.457	0.836	0.591	0.367	0.184	0.246	0.861	0.854	0.858

Best results are in bold typeface.

Table 12. Number of instances per each cluster after applying the CBUS method to the Gulf crisis training dataset

	K-means				EM			
Cluster#	Negative class	Neutral class	Positive class	Total	Negative class	Neutral class	Positive class	Total
Cluster 1	246	399	248	893	246	399	679	1324
Cluster 2	246	399	1000	1645	246	399	426	1071
Cluster 3	246	399	313	958	246	399	456	1101

Table 13. Number of instances per each cluster after applying the CBUS method to the Morocco-2016 dataset

	K-means				EM			
Cluster#	Negative class	Neutral class	Positive class	Total	Negative class	Neutral class	Positive class	Total
Cluster 1	611	368	231	1210	611	368	607	1586
Cluster 2	611	368	267	1246	611	368	394	1373
Cluster 3	611	368	849	1828	611	368	344	1323

Table 14. Number of instances per each cluster after applying the CBUS method to the LABR dataset

	K-means				EM			
Cluster#	Negative class	Neutral class	Positive class	Total	Negative class	Neutral class	Positive class	Total
Cluster 1	1543	2223	827	4593	1543	2223	2575	6341
Cluster 2	1543	2223	5592	9358	1543	2223	2368	6134
Cluster 3	1543	2223	1929	5695	1543	2223	3405	7171

Table 15. Number of instances per each cluster after applying the CBUS method to the HARD dataset

	K-means				EM			
Cluster#	Negative class	Neutral class	Positive class	Total	Negative class	Neutral class	Positive class	Total
Cluster 1	9561	14,532	14,918	39,011	9561	14,532	10,694	34,787
Cluster 2	9561	14,532	5726	29,819	9561	14,532	11,906	35,999
Cluster 3	9561	14,532	26,761	50854	9561	14,532	24,805	48,898

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 16. The evaluation metrics of the RF and the UNB classifiers using the CBUS method applied to the Gulf crisis dataset.

		Negative class			Neutral class			Positive class		
Model name	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
$RFKM_{C1}$	0.733 ± 0.008	0.734 ± 0.014	0.689 ± 0.017	0.711 ± 0.013	0.716 ± 0.010	0.739 ± 0.012	0.727 ± 0.008	0.759 ± 0.011	0.765 ± 0.017	0.762 ± 0.012
$RFKM_{C2}$	$\mathbf{0.752 \pm 0.006}$	0.696 ± 0.020	0.649 ± 0.017	0.671 ± 0.016	0.625 ± 0.021	0.371 ± 0.019	0.466 ± 0.019	0.789 ± 0.005	0.930 ± 0.006	0.853 ± 0.004
$RFKM_{C3}$	0.672 ± 0.008	0.719 ± 0.015	0.690 ± 0.015	0.704 ± 0.013	0.636 ± 0.011	0.643 ± 0.012	0.639 ± 0.009	0.684 ± 0.009	0.696 ± 0.016	0.690 ± 0.011
$NBKM_{C1}$	0.716	0.684	0.684	0.705	0.709	0.677	0.693	0.760	0.767	0.763
$NBKM_{C2}$	0.711	0.621	0.712	0.664	0.473	0.500	0.486	0.845	0.795	0.819
$NBKM_{C3}$	0.675	0.685	0.699	0.692	0.677	0.586	0.628	0.666	0.768	0.713
$RFEM_{C1}$	$\mathbf{0.743 \pm 0.007}$	0.742 ± 0.017	0.675 ± 0.017	0.707 ± 0.015	0.676 ± 0.011	0.503 ± 0.016	0.576 ± 0.014	0.768 ± 0.006	0.909 ± 0.005	0.833 ± 0.005
$RFEM_{C2}$	0.729 ± 0.008	0.739 ± 0.014	0.687 ± 0.012	0.712 ± 0.010	0.665 ± 0.011	0.678 ± 0.014	0.671 ± 0.011	0.785 ± 0.009	0.802 ± 0.012	0.793 ± 0.008
$RFEM_{C3}$	0.653 ± 0.009	0.708 ± 0.012	0.699 ± 0.023	0.704 ± 0.014	0.569 ± 0.014	0.544 ± 0.015	0.556 ± 0.012	0.693 ± 0.009	0.724 ± 0.015	0.708 ± 0.01
$NBEM_{C1}$	0.697	0.649	0.72	0.683	0.543	0.541	0.542	0.811	0.78	0.795
$NBEM_{C2}$	0.701	0.653	0.715	0.683	0.641	0.636	0.638	0.791	0.754	0.772
$NBEM_{C3}$	0.666	0.732	0.703	0.717	0.581	0.547	0.563	0.701	0.751	0.725

The best results are in bold typeface.

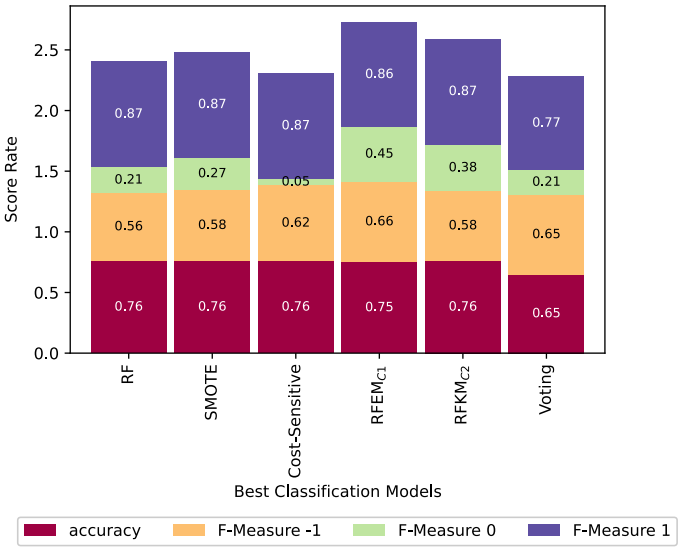

Figure 4. The evaluation metrics of the balancing algorithms, majority voting ensemble model, and the best prediction data models applied to the Gulf crisis dataset.

The reason behind selecting only two models from the 12 potential models was determined experimentally. Initially, we started building the ensemble using the best two models, four models, a combination of best models, and all twelve models, in addition to the RF and UNB models. However, the best performance of the majority voting ensemble model, shown in Table 20 and Figure 4, was achieved using the best two models combined with the RF model. The results obtained from the UNB model were unsatisfactory. Therefore, the UNB model was ignored.

2. The best two models from Table 16; RFKM$_{C2}$ and RFEM$_{C1}$ were separately applied to the imbalanced test dataset. We noticed that the RFEM$_{C1}$ model slightly outperformed the RFKM$_{C2}$ in terms of F1-score rates for both the negative and neutral classes. In addition, both models show superiority over the majority voting ensemble model in terms of accuracy and F1-score rates. Table 20 and Figure 4 show the final results of the Gulf crisis dataset.

Similarly, we repeated the same procedure on the Morocco-2016 dataset. The best data models achieving the highest accuracy rates from Table 17 were RFKM$_{C3}$ and RFEM$_{C1}$. The ensemble combined the two models with the RF model to build the final prediction from the imbalanced test dataset. The final results and the performance of the majority voting ensemble model are shown in Table 21 and Figure 5. Next, the best two models, RFKM$_{C3}$ and RFEM$_{C1}$, were separately applied to the imbalanced test dataset. We observed that the RFKM$_{C3}$ slightly outperformed the RFEM$_{C1}$. However, the majority voting ensemble model outperformed the two individual models in terms of accuracy and F1-score rates.

Whilst for the publicly available LABR dataset, the performance of the SCArD algorithm was compared to all algorithms applied to the original LABR dataset. The algorithms included SVM, MNB, and BNB (Aly and Atiya, 2013). After running the experiment on the imbalanced LABR test dataset, we noticed that the SVM algorithm achieved the highest accuracy rate of (0.674). While for the SCArD algorithm, as shown in Table 18, the best data models achieving the highest accuracy rates were the RFKM$_{C2}$ and RFKM$_{C3}$, respectively. Consequently, those two models were used in the ensemble classification step. After all, we noticed that the majority voting ensemble

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 17. The evaluation metrics of the RF and the UNB classifiers using the CBUS method applied to the Morocco-2016 dataset.

		Negative class			Neutral class			Positive class		
Model name	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
$RFKM_{C1}$	0.620 ± 0.01	0.620 ± 0.009	0.901 ± 0.01	0.734 ± 0.008	0.589 ± 0.028	0.270 ± 0.019	0.370 ± 0.021	0.658 ± 0.033	0.433 ± 0.038	0.522 ± 0.034
$RFKM_{C2}$	0.588 ± 0.009	0.576 ± 0.006	0.914 ± 0.012	0.706 ± 0.007	0.561 ± 0.028	0.251 ± 0.017	0.346 ± 0.019	0.735 ± 0.035	0.307 ± 0.024	0.433 ± 0.026
$RFKM_{C3}$	**0.679** ± 0.005	0.711 ± 0.006	0.690 ± 0.008	**0.700** ± 0.006	0.522 ± 0.009	0.568 ± 0.01	**0.544** ± 0.008	0.812 ± 0.005	0.778 ± 0.008	**0.692** ± 0.005
$NBKM_{C1}$	0.607	0.551	0.625	0.586	0.429	0.502	0.462	0.771	0.640	0.699
$NBKM_{C2}$	0.611	0.724	0.647	0.683	0.480	0.497	0.488	0.575	0.687	0.626
$NBKM_{C3}$	0.640	0.737	0.696	0.716	0.505	0.512	0.509	0.616	0.694	0.653
$RFEM_{C1}$	**0.662** $\pm$ **0.007**	0.618 ± 0.002	0.893 ± 0.01	**0.730** $\pm$ **0.006**	0.606 ± 0.034	0.257 ± 0.016	0.361 ± 0.021	0.832 ± 0.017	0.684 ± 0.014	**0.751** $\pm$ **0.011**
$RFEM_{C2}$	0.577 ± 0.008	0.564 ± 0.006	0.873 ± 0.012	0.685 ± 0.007	0.522 ± 0.04	0.211 ± 0.02	0.300 ± 0.024	0.649 ± 0.015	0.458 ± 0.016	0.537 ± 0.014
$RFEM_{C3}$	0.598 ± 0.01	0.567 ± 0.011	0.721 ± 0.016	0.635 ± 0.011	0.572 ± 0.039	0.199 ± 0.021	0.295 ± 0.027	0.638 ± 0.011	0.716 ± 0.016	0.674 ± 0.011
$NBEM_{C1}$	0.610	0.619	0.589	0.604	0.473	0.512	0.492	0.692	0.691	0.794
$NBEM_{C2}$	0.563	0.637	0.659	0.648	0.438	0.405	0.421	0.552	0.562	0.557
$NBEM_{C3}$	0.627	0.592	0.560	0.575	0.584	0.173	0.266	0.649	0.872	0.744

The best results are in bold typeface.

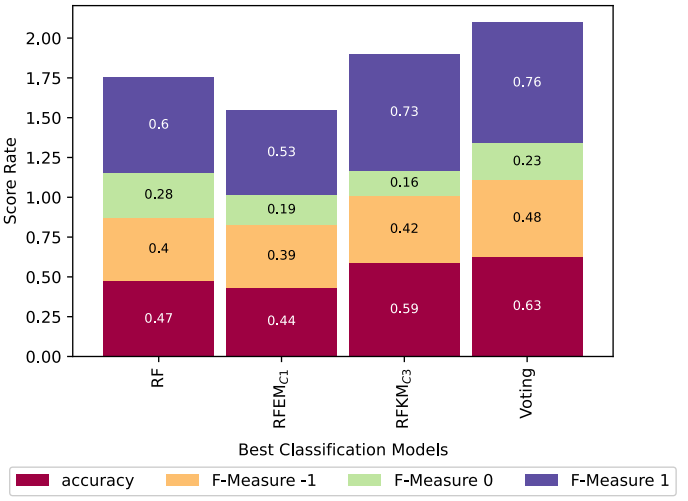

Figure 5. The evaluation metrics of the majority voting ensemble model and the best prediction data models applied to the Morocco-2016.

model outperformed all classification models in terms of accuracy and F1-score rates for negative, neutral, and positive classes. The final results and the performance of all algorithms applied to the LABR dataset are shown in Table 22 and Figure 6. Consequently, the SCArD algorithm outperformed all algorithms applied to the original LABR dataset.

Finally, for the HARD dataset, which is another publicly available dataset, we tested the performance of the SCArD algorithm and compared it to all algorithms applied to the original HARD dataset. The algorithms included Logistic Regression, AdaBoost, SVM, Passive-Aggressive, and Perceptron (Elnagar et al., 2018). After conducting the experiment on the imbalanced dataset sampled from HARD as described in Table 4, we noticed that the SVM and the Logistic Regression algorithms achieved the highest accuracy rates of (0.84). While for the SCArD algorithm, as shown in Table 19, the best data models achieving the highest accuracy rates were the RFKM$_{C1}$ and RFKM$_{C3}$, respectively. Therefore, those two models were used in the ensemble classification step. After all, we noticed that the majority voting ensemble model outperformed all classification models in terms of accuracy and F1-score rates for negative, neutral, and positive classes. The final results and the performance of all algorithms applied to the HARD dataset are shown in Table 23 and Figure 7. Accordingly, the SCArD algorithm outperformed all algorithms applied to the sampled HARD dataset.

The results of the fourth experiment indicated that using the SCArD algorithm, achieved the best F1-score rates and outperformed the other balancing algorithms applied to the four datasets. Table 24 depicts a comparison of the evaluation metrics of the four experiments on all datasets. The following behaviors were observed:

- First, for the Gulf dataset, we observed that the RFEM$_{C1}$ model significantly outperformed the conventional RF, SMOTE, and cost-sensitive models based on the F1-score rates for the minor classes. The RFEM$_{C1}$ model achieved F1-score rates of (0.663) and (0.451) for the negative and neutral classes, respectively. In comparison, the positive class achieved an F1-score rate of (0.859).
- Secondly, we observed that the proposed SCArD algorithm using the majority voting ensemble model achieved the best results for three datasets (Morocco-2016, LABR, and HARD) based on F1-score rates for the minority sentiment classes as well as for the positive

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 18. The evaluation metrics of the RF and the UNB classifiers using the CBUS method applied to the LABR dataset.

		Negative class			Neutral class			Positive class		
Model name	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
$RFKM_{C1}$	0.59 ± 0.005	0.58 ± 0.006	0.48 ± 0.012	0.52 ± 0.009	0.57 ± 0.004	0.76 ± 0.009	0.65 ± 0.004	0.79 ± 0.022	0.32 ± 0.017	0.46 ± 0.02
$RFKM_{C2}$	$\mathbf{0.71 \pm 0.002}$	0.62 ± 0.009	0.35 ± 0.007	0.45 ± 0.007	0.59 ± 0.008	0.34 ± 0.007	0.43 ± 0.007	0.74 ± 0.001	0.95 ± 0.002	0.83 ± 0.001
$RFKM_{C3}$	$\mathbf{0.66 \pm 0.003}$	0.61 ± 0.009	0.44 ± 0.008	0.51 ± 0.008	0.63 ± 0.005	0.62 ± 0.006	0.63 ± 0.005	0.7 ± 0.003	0.87 ± 0.005	0.78 ± 0.003
$NBKM_{C1}$	0.54	0.58	0.45	0.51	0.71	0.43	0.53	0.40	1.00	0.57
$NBKM_{C2}$	0.56	0.40	0.39	0.39	0.35	0.53	0.42	0.78	0.63	0.69
$NBKM_{C3}$	0.62	0.52	0.51	0.51	0.61	0.52	0.57	0.69	0.82	0.75
$RFEM_{C1}$	0.63 ± 0.003	0.58 ± 0.005	0.44 ± 0.011	0.5 ± 0.008	0.54 ± 0.006	0.6 ± 0.007	0.57 ± 0.005	0.73 ± 0.003	0.76 ± 0.005	0.74 ± 0.003
$RFEM_{C2}$	0.58 ± 0.002	0.53 ± 0.005	0.5 ± 0.004	0.51 ± 0.004	0.52 ± 0.003	0.56 ± 0.003	0.54 ± 0.003	0.67 ± 0.002	0.65 ± 0.003	0.66 ± 0.002
$RFEM_{C3}$	0.65 ± 0.002	0.59 ± 0.005	0.36 ± 0.004	0.45 ± 0.005	0.58 ± 0.002	0.47 ± 0.003	0.52 ± 0.002	0.69 ± 0.001	0.9 ± 0.002	0.78 ± 0.001
$NBEM_{C1}$	0.61	0.56	0.43	0.49	0.65	0.34	0.45	0.61	0.94	0.74
$NBEM_{C2}$	0.52	0.49	0.45	0.47	0.47	0.49	0.48	0.59	0.58	0.59
$NBEM_{C3}$	0.57	0.45	0.39	0.42	0.46	0.60	0.52	0.73	0.63	0.68

The best results are in bold typeface.

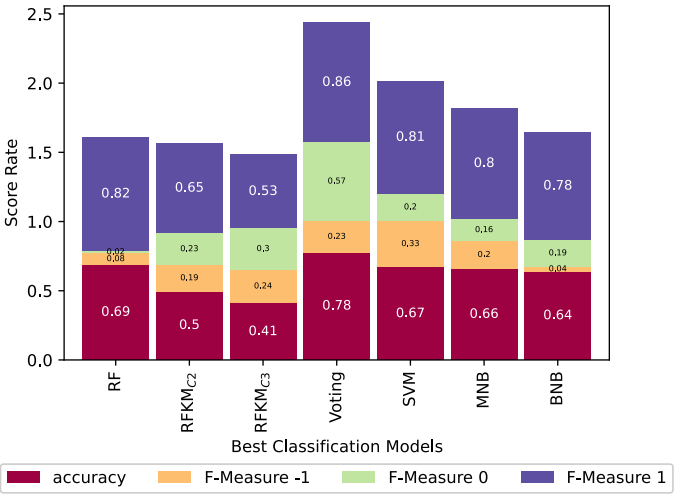

Figure 6. The evaluation metrics of the majority voting ensemble model and the best prediction data models applied to the LABR.

class. For instance, in the Morocco-2016 dataset, shown in Table 24, the voting ensemble achieved F1-score rates of (0.48) and (0.23) for the negative and neutral sentiments classes, while it achieved an F1-score rate of (0.76) for the positive class. While for the LABR dataset, the SCArD voting ensemble outperformed all classification models applied to the dataset, as described in Table 22. Table 24 shows that the voting ensemble applied to the LABR dataset achieved F1-score rates of (0.230) and (0.57) for the negative and neutral classes, respectively, and (0.860) for the positive class. On the other hand, for the HARD dataset, the SCArD voting ensemble also outperformed all classification models applied to the dataset, as described in Table 23. Table 24 shows that the voting ensemble applied to the HARD dataset achieved F1-score rates of (0.730) and (0.80) for the negative and neutral classes, respectively, and (0.930) for the positive class.

Intuitively, this was expected because the SCArD algorithm did not remove the majority of class instances from the dataset. Yet, it did not inject the dataset with artificial minority class instances or change the class distribution.

4.3.5 Experiment V: Classification with the SCArD algorithm: Comparing TF-IDF and Arabic-BERT pre-trained language model for feature extraction

For tasks that require a deep understanding of the language, BERT, in general, is a more complex and powerful technique for feature extraction compared to TF-IDF. However, TF-IDF remains a simple, useful, and computationally efficient technique for tasks that require a simple bag-of-words representation of the text. Training Arabic-BERT from scratch on a new text dataset requires powerful computing resources; usually, a cloud TPU or GPUs are used, which is extremely expensive for academic research groups. A way around this problem was to extract vectors directly from the Arabic-BERT model. Therefore, in this experiment, we used the knowledge learned from the Arabic-BERT model to encode the input text from each of the LABR and HARD datasets into a fixed-length vector representation, which captures the contextual meaning of the input text as described by Algorithm 1. This approach can save significant time and resources compared to training the model from scratch.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 19. The evaluation metrics of the RF and the UNB classifiers using the CBUS method applied to the HARD dataset.

		Negative class			Neutral class			Positive class		
Model name	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
RFKM$_{C1}$	**0.82 ± 0.001**	0.8 ± 0.003	0.67 ± 0.003	0.73 ± 0.002	0.8 ± 0.002	0.79 ± 0.002	0.79 ± 0.002	0.84 ± 0.002	0.95 ± 0.001	0.9 ± 0.001
RFKM$_{C2}$	0.8 ± 0.003	0.8 ± 0.005	0.69 ± 0.003	0.74 ± 0.003	0.79 ± 0.003	0.86 ± 0.003	0.82 ± 0.002	0.86 ± 0.003	0.84 ± 0.007	0.85 ± 0.003
RFKM$_{C3}$	**0.83 ± 0.001**	0.79 ± 0.003	0.65 ± 0.004	0.71 ± 0.002	0.79 ± 0.002	0.7 ± 0.003	0.74 ± 0.002	0.85 ± 0.001	0.96 ± 0.001	0.9 ± 0.001
NBKM$_{C1}$	0.81	0.75	0.72	0.73	0.79	0.78	0.79	0.86	0.90	0.88
NBKM$_{C2}$	0.80	0.77	0.73	0.75	0.82	0.82	0.82	0.82	0.88	0.85
NBKM$_{C3}$	0.76	0.69	0.73	0.71	0.63	0.76	0.69	0.89	0.77	0.83
RFEM$_{C1}$	0.75 ± 0.002	0.77 ± 0.003	0.68 ± 0.003	0.72 ± 0.003	0.7 ± 0.002	0.82 ± 0.003	0.75 ± 0.002	0.81 ± 0.003	0.7 ± 0.003	0.75 ± 0.003
RFEM$_{C2}$	0.74 ± 0.002	0.76 ± 0.002	0.71 ± 0.004	0.73 ± 0.002	0.7 ± 0.004	0.75 ± 0.002	0.73 ± 0.002	0.77 ± 0.002	0.74 ± 0.004	0.75 ± 0.002
RFEM$_{C3}$	0.82 ± 0.001	0.79 ± 0.003	0.62 ± 0.004	0.7 ± 0.002	0.78 ± 0.003	0.7 ± 0.002	0.74 ± 0.002	0.84 ± 0.001	0.96 ± 0.001	0.9 ± 0.001
NBEM$_{C1}$	0.70	0.81	0.60	0.69	0.80	0.62	0.70	0.59	0.91	0.72
NBEM$_{C2}$	0.72	0.72	0.73	0.73	0.72	0.70	0.71	0.71	0.72	0.72
NBEM$_{C3}$	0.75	0.71	0.73	0.72	0.62	0.77	0.69	0.88	0.76	0.82

The best results are in bold typeface.

Table 20. The evaluation metrics of the RF classifier, balancing algorithms, majority voting ensemble, and the best prediction data models applied to the Gulf crisis test dataset.

		Negative class			Neutral class			Positive class		
Model	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
RF	0.762	0.789	0.445	0.560	0.464	0.140	0.213	0.779	0.978	0.867
SMOTE	0.763	0.755	0.474	0.581	0.456	0.192	0.270	0.793	0.961	0.869
Cost-Sensitive	0.762	0.667	0.589	0.624	0.505	0.028	0.053	0.777	0.984	0.868
$RFEM_{C1}$	**0.753**	0.753	0.592	**0.663**	0.406	0.506	**0.451**	0.871	0.846	**0.859**
$RFKM_{C2}$	0.760	0.762	0.466	0.578	0.439	0.342	0.384	0.819	0.922	0.867
Voting	0.652	0.726	0.592	0.652	0.200	0.215	0.207	0.769	0.778	0.773

The best results are in bold typeface.

Table 21. The evaluation metrics of the RF classifier, majority voting ensemble, and the best prediction data models applied to the Morocco-2016 test dataset.

		Negative class			Neutral class			Positive class		
Model	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
RF	0.474	0.38	0.430	0.400	0.210	0.400	0.280	0.730	0.510	0.600
$RFEM_{C1}$	0.436	0.3	0.580	0.390	0.250	0.150	0.190	0.660	0.450	0.530
$RFKM_{C3}$	0.589	0.41	0.43	0.42	0.31	0.11	0.16	0.69	0.79	0.73
Voting	**0.629**	0.46	0.51	**0.48**	0.43	0.16	**0.23**	0.72	0.81	**0.76**

The best results are in bold typeface.

Table 22. The evaluation metrics of the RF classifier, majority voting ensemble, and the best prediction data models applied to the LABR test dataset.

		Negative class			Neutral class			Positive class		
Model	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
RF	0.690	0.510	0.050	0.080	0.170	0.010	0.020	0.700	0.980	0.820
$RFKM_{C2}$	0.497	0.260	0.150	0.190	0.180	0.320	0.230	0.690	0.610	0.650
$RFKM_{C3}$	0.413	0.330	0.19	0.240	0.200	0.640	0.300	0.780	0.400	0.530
Voting	**0.776**	0.790	0.140	**0.230**	0.900	0.410	**0.570**	0.760	0.990	**0.860**
SVM	0.674	0.500	0.240	0.330	0.260	0.160	0.200	0.740	0.890	0.810
MNB	0.660	0.400	0.130	0.200	0.230	0.130	0.160	0.720	0.890	0.8
BNB	0.637	0.260	0.020	0.040	0.210	0.160	0.190	0.710	0.870	0.780

The best results are in bold typeface.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 23. The evaluation metrics of the RF classifier, majority voting ensemble, and the best prediction data models applied to the HARD test dataset.

		Negative class			Neutral class			Positive class		
Model	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
RF	0.826	0.880	0.500	0.640	0.880	0.500	0.640	0.810	0.990	0.890
$RFKM_{C1}$	0.547	0.690	0.610	0.650	0.300	0.800	0.440	0.880	0.460	0.600
$RFKM_{C3}$	0.676	0.700	0.61	0.650	0.390	0.710	0.510	0.870	0.680	0.760
Voting	**0.881**	0.910	0.610	**0.730**	0.920	0.710	**0.800**	0.870	0.990	**0.930**
Logistic Regression	0.843	0.760	0.670	0.710	0.780	0.590	0.670	0.870	0.950	0.910
AdaBoost	0.826	0.880	0.500	0.640	0.880	0.500	0.640	0.810	0.990	0.89
SVM	0.844	0.760	0.690	0.720	0.780	0.570	0.660	0.870	0.960	0.910
Passive Aggressive	0.791	0.660	0.620	0.640	0.630	0.550	0.590	0.860	0.910	0.880
Perceptron	0.784	0.610	0.620	0.620	0.610	0.560	0.580	0.870	0.890	0.880

The best results are in bold typeface.

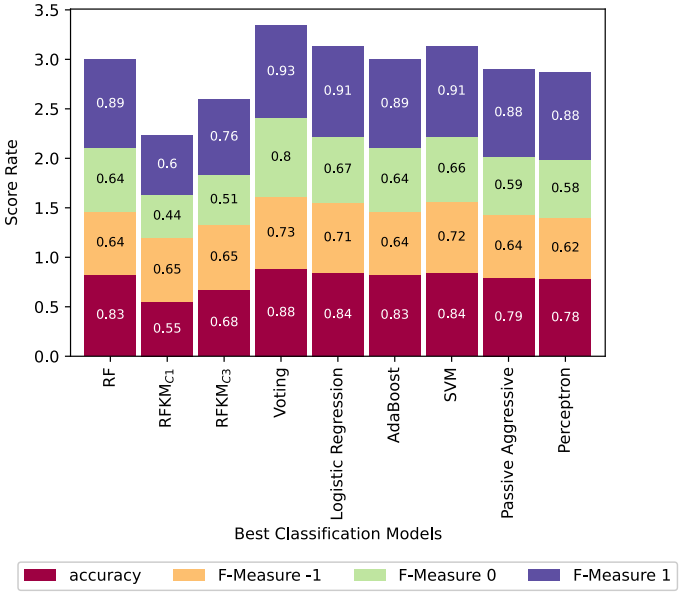

Figure 7. The evaluation metrics of the majority voting ensemble model and the best prediction data models applied to the HARD.

The results of using the Arabic-BERT data model for feature extraction are in Table 25. As shown from this table, using the SCArD voting ensemble with the Arabic-BERT pretrained model for feature extraction achieved the best F1-score rates for both datasets, LABR and HARD. As expected, the Arabic-BERT model should outperform the TF-IDF statistical model for feature extraction. However, the SCArD voting ensemble gave its best F1-score results using the TF-IDF model, as discussed earlier in experiment IV and shown in Table 24. It is usually not uncommon

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Table 24. Summarization of the best classification models applied to all datasets based on the F1-score rates for negative (−ve), neutral (N), and positive (+ve) sentiment classes

Experiment #	Dataset/Language model	Best classification model	Accuracy	−ve	N	+ve
I	Gulf	RF (FS4)	0.762	0.567	0.213	0.867
II	Gulf	RF (FS4) + SMOTE	0.763	0.581	0.270	0.869
III	Gulf	RF (FS4) + Cost-sensitive	0.762	0.624	0.053	0.869
IV	Gulf	SCArD($RFEM_{C1}$)	0.753	0.663	0.451	0.859
IV	Morocco-2016	SCArD(Voting)	0.629	0.480	0.230	0.760
IV	LABR	SCArD(Voting)	0.776	0.230	0.570	0.860
IV	HARD	SCArD(Voting)	0.881	0.730	0.800	0.930

Table 25. Summarization of the best models of the LABR and HARD datasets trained on Arabic-BERT based on the F1-score rates for negative (−ve), neutral (N), and positive (+ve) sentiment classes

Dataset	Model	Accuracy	(−ve)	(N)	(+ve)
LABR	RF	0.695	0.01	0.02	0.82
	$RFEM_{C1}$	0.478	0.15	0.24	0.64
	$RFKM_{C2}$	0.478	0.16	0.24	0.64
	Voting	0.531	0.10	0.22	0.69
HARD	RF	0.721	0.28	0.31	0.83
	$RFKM_{C1}$	0.44	0.41	0.39	0.50
	$RFKM_{C2}$	0.404	0.44	0.37	0.43
	Voting	0.592	0.42	0.42	0.71

to observe that the performance of BERT and its variances, such as the Arabic-BERT, might be affected based on the similarity of the new datasets to the original pre-training data. Here are two main reasons why the results of using the Arabic-BERT model for feature extraction did not perform well compared to TF-IDF on the two datasets:

1. The LABR and HARD datasets contain words that might not be present in the original Arabic-BERT pretrained model. Therefore, the model might have difficulty understanding these out-of-vocabulary words as it may not have learned their representations.
2. The domain of the new datasets significantly differs from the data in the original Arabic-BERT pretrained model; however, fine-tuning the model on the new datasets to adapt their language representations becomes necessary to achieve good performance.

For future research, we plan to train the Arabic-BERT model on all datasets used in this research.

5. Conclusions and future directions

The main task of text data mining is to extract hidden knowledge from text using techniques borrowed from NLP and data mining fields. In this study, we tackled the problem of Arabic

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

multi-class sentiment classification. We implemented several models to address the issue of imbalanced datasets. We also presented their impacts on the classification problem. The experiments conducted on the Gulf crisis dataset showed that selecting the title and the source of an article decreased the required features. Consequently, they increased the performance of the classification measures. After analyzing the results of the experiments, we concluded that the cost-sensitive classifier, which depends on a cost matrix to handle the imbalanced datasets, performed better than the SMOTE oversampling method. The clustered-based Undersampling method, incorporated in the proposed clustering approach, balanced the ratio between the major and the minor classes. This was achieved by decreasing the number of major class instances and maintaining the number of minor class instances at the cluster level. Also, we concluded that the clustered-based undersampling method outperformed the other tested models. We applied two techniques to test our approach and generate new predictions from an imbalanced test dataset: (1) Using the best prediction data models or (2) Using the majority voting ensemble model, which combines the best prediction data models to generate the final predictions. The best prediction data models effectively outperformed the majority voting ensemble model for the Gulf crisis dataset. However, the majority voting ensemble model performed better for the Morocco-2016 dataset and the publicly available datasets, LABR and HARD. Finally, we experienced two feature extraction schemes; the statistical TF-IDF and the Arabic-BERT pre-trained language model. Arabic-BERT is a pretrained language model that can generate high-quality numerical embeddings of text data. Although we expected the Arabic-BERT model to have superiority over TF-IDF, however, the results were in favor of TF-IDF. The main reason behind this behavior was that the Arabic-BERT model was not appropriately trained on the test datasets. For future work, more investigations of the proposed approach will be considered. This would include carrying out the following tasks:

- Concentrating on the behavior of the instances in the minority classes to better understand the main learning difficulties.
- Applying the proposed algorithm to other sentiment classification platforms like Twitter.
- Introducing other clustering algorithms that could enhance the sentiment classification results.
- Training the Arabic-BERT model from scratch for better performance.

Acknowledgments. The authors would like to thank Eng. Khaled Taha, CEO & Founder of Toot for Media Solutions & Services, kt@tootvs.com for permission to use the Gulf crisis conflict and the Morocco-2016 datasets. Also, we would like to thank Toot's experts for annotating and evaluating the Gulf crisis dataset. The work of Bassam Hammo was on sabbatical leave from 2021 to 2022 from the King Abdullah II School of Information Technology, The University of Jordan, to the Department of Software Engineering, King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan.

Funding. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

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

Abbasi A., Chen H. and Salem A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems* **26**(3), 34.

Abdulla N. A., Ahmed N. A., Shehab M. A. and Al-Ayyoub M. (2013). *Arabic sentiment analysis: Lexicon-based and corpus-based*. In Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference, IEEE, pp. 1–6.

Aftab E. and Malik M. K. (2022). eRock at Qur'an QA 2022: Contemporary deep neural networks for Qur'an based reading comprehension question answers. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools*

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, Marseille, France: European Language Resources Association, pp. 96–103.

Ahmad M., Aftab S., Bashir M. S., Hameed N., Ali I. and Nawaz Z. (2018). SVM optimization for sentiment analysis. *International Journal of Advanced Computer Science and Applications* **9**(4), 393–398.

Al-Azani S. and El-Alfy E.-S. M. (2017). Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short Arabic text. *Procedia Computer Science* **109**, 359–366.

Al-Laith A. and Shahbaz M. (2021). Tracking sentiment towards news entities from Arabic news on social media. *Future Generation Computer Systems* **118**, 467–484.

Al-Moslmi T., Albared M., Al-Shabi A., Omar N. and Abdullah S. (2018). Arabic senti-lexicon: Constructing publicly available language resources for Arabic sentiment analysis. *Journal of Information Science* **44**(3), 345–362.

Al-Sughaiyer I. A. and Al-Kharashi I. A. (2004). Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology* **55**(3), 189–213.

Alayba A. M., Palade V., England M. and Iqbal R. (2018). *A combined CNN and LSTM model for Arabic sentiment analysis*. In International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Springer, pp. 179–191.

AlMahmoud R. H., Hammo B. and Faris H. (2020). A modified bond energy algorithm with fuzzy merging and its application to Arabic text document clustering. *Expert Systems with Applications* **159**, 113598.

Almas Y. and Ahmad K. (2007). A note on extracting 'sentiments' in financial news in English, Arabic & Urdu, *The Second Workshop on Computational Approaches to Arabic Script-Based Languages*, pp. 1–12.

Alrefai M., Faris H. and Aljarah I. (2018). Sentiment analysis for Arabic language: A brief survey of approaches and techniques, arXiv preprint arXiv: 1809.02782.

Aly M. and Atiya A. (2013). *LABR: A large scale Arabic book reviews dataset*. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Sofia, Bulgaria: Association for Computational Linguistics, pp. 494–498.

Amrit C., Paauw T., Aly R. and Lavric M. (2017). Identifying child abuse through text mining and machine learning. *Expert Systems with Applications* **88**, 402–418.

Andrews N. O. and Fox E. A. (2007). *Recent developments in document clustering*. Technical Report TR-07-35, Computer Science, Virginia Tech.

Araque O., Corcuera-Platas I., SáNchez-Rada J. F. and Iglesias C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications* **77**, 236–246.

Aref A., Al Mahmoud R. H., Taha K. and Al-Sharif M. (2020). Hate speech detection of Arabic short text. *Computer Science and Information Technology* **10**, 81–94.

Assiri A., Emam A. and Al-Dossari H. (2018). Towards enhancement of a lexicon-based approach for Saudi dialect sentiment analysis. *Journal of Information Science* **44**(2), 184–202.

Aung K. Z. and Myo N. N. (2017). *Sentiment analysis of students' comment using lexicon based approach*. In 2017 IEEE/ACIS 16th International Conference on Computer and Information p-Science (ICIS), IEEE, pp. 149–154.

Batista G. E., Prati R. C. and Monard M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter* **6**(1), 20–29.

Bayoudhi A., Ghorbel H., Koubaa H. and Belguith L. H. (2015). Sentiment classification at discourse segment level: Experiments on multi-domain Arabic corpus. *Journal for Language Technology and Computational Linguistics* **30**(1), 1–24.

Bekkar M. and Alitouche T. A. (2013). Imbalanced data learning approaches review. *International Journal of Data Mining & Knowledge Management Process* **3**(4), 15.

Biltawi M., Etaiwi W., Tedmori S., Hudaib A. and Awajan A. (2016). *Sentiment classification techniques for Arabic language: A survey*. In Information and Communication Systems (ICICS), 2016 7th International Conference, IEEE, pp. 339–346.

Bonta V. and Janardhan N. K. N. (2019). A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology* **8**(S2), 1–6.

Boudlal A., Lakhouaja A., Mazroui A., Meziane A., Bebah M. and Shoul M. (2010). *Alkhalil morpho sys1: A morphosyntactic analysis system for Arabic texts*. In International Arab Conference on Information Technology, Benghazi, Libya, pp. 1–6.

Breiman L. (2001). Random forests. *Machine Learning* **45**(1), 5–32.

Chang J.-R., Chen L.-S. and Lin L.-W. (2021). A novel cluster based over-sampling approach for classifying imbalanced sentiment data. *IAENG International Journal of Computer Science* **48**(4), 1118–1128.

Charbuty B. and Abdulazeez A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends* **2**(01), 20–28.

Chawla N. V., Bowyer K. W., Hall L. O. and Kegelmeyer W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* **16**, 321–357.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805.

Dhillon I. S., Mallela S. and Kumar R. (2003). A divisive information theoretic feature clustering algorithm for text classification. *The Journal of Machine Learning Research* **3**, 1265–1287.

Drummond C. and Holte R. C. (2003). C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling, *Workshop on Learning from Imbalanced Datasets II*, **11**, Citeseer, pp. 1–8.

El-Affendi M. A., **Alrajhi K. and Hussain A.** (2021). A novel deep learning-based multilevel parallel attention neural (MPAN) model for multidomain Arabic sentiment analysis. *IEEE Access* **9**, 7508–7518.

El-Halees A. M. (2011). *Arabic opinion mining using combined classification approach*. In International Arab Conference on Information Technology, Naif Arab University for Security Sciences, pp. 1–6.

Elfaik H. and et al. (2021). Deep bidirectional LSTM network learning-based sentiment analysis for Arabic text. *Journal of Intelligent Systems* **30**(1), 395–412.

Elkan C. (2001). *The foundations of cost-sensitive learning*. In International Joint Conference on Artificial Intelligence, Lawrence Erlbaum Associates Ltd, vol **17**, pp. 973–978.

Elnagar A., **Khalifa Y. S. and Einea A.** (2018). Hotel Arabic-reviews dataset construction for sentiment analysis applications. In: Shaalan, K., Hassanien, A., Tolba, F. (eds) *Intelligent Natural Language Processing: Trends and Applications*, Studies in Computational Intelligence, vol 740. Springer, Cham. https://doi.org/10.1007/978-3-319-67056-0_3

Emami J., **Nugues P.**, **Elnagar A. and Afyouni I.** (2022). *Arabic image captioning using pre-training of deep bidirectional transformers*. In Proceedings of the 15th International Conference on Natural Language Generation, pp. 40–51.

Farra N., **Challita E.**, **Assi R. A. and Hajj H.** (2010). *Sentence-level and document-level sentiment mining for Arabic texts*. In Data Mining Workshops (ICDMW), 2010 IEEE International Conference, IEEE, pp. 1114–1119.

Fernández A., **García S.**, **Galar M.**, **Prati R. C.**, **Krawczyk B. and Herrera**, **F.** (2018). Performance measures. In: *Learning from Imbalanced Data Sets*, Springer, Cham, pp. 47–61. https://doi.org/10.1007/978-3-319-98074-4_3

Ganganwar V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering* **2**(4), 42–47.

George S. and Srividhya V. (2022). Performance evaluation of sentiment analysis on balanced and imbalanced dataset using ensemble approach. *Indian Journal of Science and Technology* **15**(17), 790–797.

Ghosh K., **Banerjee A.**, **Chatterjee S. and Sen S.** (2019). *Imbalanced twitter sentiment analysis using minority oversampling*. In 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), IEEE, pp. 1–5.

Gupta V. and Lehal G. S. (2010). A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence* **2**(3), 258–268.

Hammo B., **Abuleil S.**, **Lytinen S. and Evens M.** (2004). Experimenting with a question answering system for the Arabic language. *Computers and the Humanities* **38**(4), 397–415.

Hammo B., **Yagi S.**, **Ismail O. and AbuShariah M.** (2016). Exploring and exploiting a historical corpus for Arabic. *Language Resources and Evaluation* **50**(4), 839–861.

Hammo B. H. (2009). Towards enhancing retrieval effectiveness of search engines for diacritisized Arabic documents. *Information Retrieval* **12**(3), 300–323.

Hartmann J., **Huppertz J.**, **Schamp C. and Heitmann M.** (2019). Comparing automated text classification methods. *International Journal of Research in Marketing* **36**(1), 20–38.

He H., **Bai Y.**, **Garcia E. A. and Li S.** (2008). *Adasyn: Adaptive synthetic sampling approach for imbalanced learning*. In Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference, IEEE, pp. 1322–1328.

He H. and Garcia E. A. (2008). Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering* **21**, 1263–1284.

Huang A. (2008). *Similarity measures for text document clustering*. In Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand, **4**, pp. 9–56.

Hussein D. M. E.-D. M. (2018). A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences* **30**(4), 330–338.

Imran A. S., **Yang R.**, **Kastrati Z.**, **Daudpota S. M. and Shaikh S.** (2022). The impact of synthetic text generation for sentiment analysis using GAN based models. *Egyptian Informatics Journal* **23**(3), 547–557.

Janani R. and Vijayarani S. (2019). Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications* **134**, 192–200.

Jiang W., **Zhou K.**, **Xiong C.**, **Du G.**, **Ou C. and Zhang J.** (2022). KSCB: A novel unsupervised method for text sentiment analysis. *Applied Intelligence*, **53**(1), 301–311.

Jing L. (2008). *Survey of Text Clustering*. HongKong, China: Department of Mathematics, The University of Hong Kong, pp. 7695–1754.

Jing L.-P., **Huang H.-K. and Shi H.-B.** (2002). *Improved feature selection approach tfidf in text mining*. In Machine Learning and Cybernetics, 2002. Proceedings of 2002 International Conference, IEEE, vol **2**, pp. 944–946.

Kadhim A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review* **52**(1), 273–292.

Kaji N. and Kitsuregawa M. (2007). *Building lexicon for sentiment analysis from massive collection of HTML documents*. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic: Association for Computational Linguistics, pp. 1075–1083.

Khanday A. M. U. D., **Rabani S. T.**, **Khan Q. R.**, **Rouf N. and Mohi Ud Din M.** (2020). Machine learning based approaches for detecting Covid-19 using clinical text data. *International Journal of Information Technology* **12**(3), 731–739.

Khoja S. and Garside R. (1999). *Stemming Arabic Text*. Lancaster, UK: Computing Department, Lancaster University.

Khoo C. S. and Johnkhan S. B. (2018). Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. *Journal of Information Science* **44**(4), 491–511.

Kim D., **Koo J. and Kim U.-M.** (2021). *Envbert: multi-label text classification for imbalanced, noisy environmental news data*. In 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM), IEEE, pp. 1–8.

Kolchyna O., **Souza T. T.**, **Treleaven P. and Aste T.** (2015). Twitter sentiment analysis: Lexicon method, machine learning method and their combination, arXiv preprint arXiv: 1507.00955.

Koppel M. and Schler J. (2006). The importance of neutral examples for learning sentiment. *Computational Intelligence* **22**, 100–109.

Kumar M. and Sheshadri H. (2012). On the classification of imbalanced datasets. *International Journal of Computer Applications* **44**(8), 1–7.

Kyriakopoulou A. and Kalamboukis T. (2006). *Text classification using clustering*. In Proceedings of the Discovery Challenge Workshop at ECML/PKDD 2006, pp. 28–38.

Larkey L. S., **Ballesteros L. and Connell M. E.** (2002). *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, Association for Computing Machinery, pp. 275–282.

Li S., **Zhou G.**, **Wang Z.**, **Lee S. Y. M. and Wang R.** (2011). *Imbalanced sentiment classification*. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 2469–2472.

Li Y., **Guo H.**, **Zhang Q.**, **Gu M. and Yang J.** (2018). Imbalanced text sentiment classification using universal and domain-specific knowledge. *Knowledge-Based Systems* **160**, 1–15.

Liu X.-Y., **Wu J. and Zhou Z.-H.** (2008). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **39**(2), 539–550.

Madabushi H. T., **Kochkina E. and Castelle M.** (2020). Cost-sensitive Bert for generalisable sentence classification with imbalanced data, arXiv preprint arXiv: 2003.11563.

Medhat W., **Hassan A. and Korashy H.** (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* **5**(4), 1093–1113.

Mir N. M., **Khan S.**, **Butt M. A. and Zaman M.** (2016). An experimental evaluation of Bayesian classifiers applied to intrusion detection. *Indian Journal of Science and Technology* **9**(12), 1–7.

Mountassir A., **Benbrahim H. and Berrada I.** (2012). *An empirical study to address the problem of unbalanced data sets in sentiment classification*. In 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, pp. 3298–3303.

Mukhtar N., **Khan M. A. and Chiragh N.** (2018). Lexicon-based approach outperforms supervised machine learning approach for Urdu sentiment analysis in multiple domains. *Telematics and Informatics* **35**(8), 2173–2183.

Onan A. (2017). Hybrid supervised clustering based ensemble scheme for text classification. *Kybernetes*, **46**(2), 330–348.

Oueslati O., **Cambria E.**, **HajHmida M. B. and Ounelli H.** (2020). A review of sentiment analysis research in Arabic language. *Future Generation Computer Systems* **112**, 408–430.

Pang B., **Lee L. and Vaithyanathan S.** (2002). *Thumbs up?: sentiment classification using machine learning techniques*. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, vol **10**, pp. 79–86.

Pasha A., **Al-Badrashiny M.**, **Diab M. T.**, **El Kholy A.**, **Eskander R.**, **Habash N.**, **Pooleery M.**, **Rambow O. and Roth R.** (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic, *LREC*, **14**, pp. 1094–1101.

Ramyachitra D. and Manikandan P. (2014). Imbalanced dataset classification and solutions: A review. *International Journal of Computing and Business Research (IJCBR)* **5**(4), 1–29.

Ridgeway G., **Madigan D.**, **Richardson T. and O'Kane J.** (1998). Interpretable boosted Naïve Bayes classification, *KDD*, pp. 101–104.

Rojarath A., **Songpan W. and Pong-inwong C.** (2016). *Improved ensemble learning for classification techniques based on majority voting*. In 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), IEEE, pp. 107–110.

Roul R. K., **Gugnani S. and Kalpeshbhai S. M.** (2015). *Clustering based feature selection using extreme learning machines for text classification*. In 2015 Annual IEEE India Conference (INDICON), IEEE, pp. 1–6.

Roy S. S., **Dey S. and Chatterjee S.** (2020). Autocorrelation aided random forest classifier-based bearing fault detection framework. *IEEE Sensors Journal* **20**(18), 10792–10800.

Rupapara V., **Rustam F.**, **Shahzad H. F.**, **Mehmood A.**, **Ashraf I. and Choi G. S.** (2021). Impact of smote on imbalanced text features for toxic comments classification using RVVC model. *IEEE Access* **9**, 78621–78634.

Sadegh M., **Ibrahim R. and Othman Z. A.** (2012). Opinion mining and sentiment analysis: A survey. *International Journal of Computers & Technology* **2**(3), 171–178.

https://doi.org/10.1017/S135132492300027X Published online by Cambridge University Press

Safaya A., Abdullatif M. and Yuret D. (2020). *KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media*. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona: International Committee for Computational Linguistics, pp. 2054–2059.

Satriaji W. and Kusumaningrum R. (2018). *Effect of synthetic minority oversampling technique (smote), feature representation, and classification algorithm on imbalanced sentiment analysis*. In 2018 2nd International Conference on Informatics and Computational Sciences (ICICoS), IEEE, pp. 1–5.

Shaikh S., Daudpota S. M., Imran A. S. and Kastrati Z. (2021). Towards improved classification accuracy on highly imbalanced text dataset using deep neural language models. *Applied Sciences* **11**(2), 869.

Shayaa S., Jaafar N. I., Bahri S., Sulaiman A., Wai P. S., Chung Y. W., Piprani A. Z. and Al-Garadi M. A. (2018). Sentiment analysis of big data: Methods, applications, and open challenges. *IEEE Access* **6**, 37807–37827.

Shoukry A. and Rafea A. (2012). *Sentence-level Arabic sentiment analysis*. In Collaboration Technologies and Systems (CTS), 2012 International Conference, IEEE, pp. 546–550.

Singh J., Singh G. and Singh R. (2017). Optimization of sentiment analysis using machine learning classifiers. *Human-Centric Computing and Information Sciences* **7**(1), 1–12.

Singh V. K., Tiwari N. and Garg S. (2011). *Document clustering using k-means, heuristic k-means and fuzzy c-means*. In 2011 International Conference on Computational Intelligence and Communication Networks, IEEE, pp. 297–301.

Sokolova M. and Lapalme G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management* **45**(4), 427–437.

Su Y., Zhang Y., Ji D., Wang Y. and Wu H. (2012). Ensemble learning for sentiment classification, *Workshop on Chinese Lexical Semantics*. Springer, pp. 84–93.

Sun Y., Kamel M. S., Wong A. K. and Wang Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* **40**(12), 3358–3378.

Taboada M., Brooke J., Tofiloski M., Voll K. and Stede M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics* **37**(2), 267–307.

Taha A. (2017). Hybrid approach for sentiment analysis of Arabic tweets based on deep learning model and features weighting. *International Journal of Advanced and Applied Sciences* **4**, 43–49.

Tang J., Alelyani S. and Liu H. (2014). Feature selection for classification: A review. In Data Classification: Algorithms and Applications. CRC Press, pp. 37–64. https://doi.org/10.1201/b17320

Tedmori S. and Awajan A. (2019). Sentiment analysis main tasks and applications: A survey. *Journal of Information Processing Systems* **15**(3), 500–519.

Turney P. D. (2002). *Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews*. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 417–424.

Verma B. and Thakur R. S. (2018). *Sentiment analysis using lexicon and machine learning-based approaches: A survey*. In Proceedings of International Conference on Recent Advancement on Computer and Communication, Springer, pp. 441–447.

Vinodhini G. and Chandrasekaran R. (2012). Sentiment analysis and opinion mining: A survey. *International Journal* **2**(6), 282–292.

Weiss G. M. (2004). Mining with rarity: A unifying framework. *ACM SIGKDD Explorations Newsletter* **6**(1), 7–19.

Whitehead M. and Yaeger L. (2010). Sentiment mining using ensemble classification models, *Innovations and Advances in Computer Sciences and Engineering*. Springer, pp. 509–514.

Xia R., Zong C. and Li S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences* **181**(6), 1138–1152.

Xu D. and Tian Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science* **2**(2), 165–193.

Xu G., Meng Y., Qiu X., Yu Z. and Wu X. (2019). Sentiment analysis of comment texts based on BiLSTM. *IEEE Access* **7**, 51522–51532.

Yang P. and Chen Y. (2017). *A survey on sentiment analysis by using machine learning methods*. In 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), IEEE, pp. 117–121.

Yong Z., Youwen L. and Shixiong X. (2009). An improved KNN text classification algorithm based on clustering. *Journal of Computers* **4**(3), 230–237.

Zhang L., Ghosh R., Dekhil M., Hsu M. and Liu B. (2011). Combining lexicon-based and learning-based methods for twitter sentiment analysis, HP Laboratories, Technical Report HPL-2011, 89.

Cite this article: Al Mahmoud RH, Hammo BH and Faris H (2024). Cluster-based ensemble learning model for improving sentiment classification of Arabic documents. *Natural Language Engineering* **30**, 1091–1129. https://doi.org/10.1017/S135132492300027X