IEEE Open Journal of the Computer Society

Received 20 May 2024; revised 2 September 2024; accepted 5 October 2024. Date of publication 8 October 2024; date of current version 21 October 2024. The review of this article was arranged by Associate Editor Nandha Kumar Thulasiraman.

Digital Object Identifier 10.1109/OJCS.2024.3476378

Polarity Classification of Low Resource Roman Urdu and Movie Reviews Sentiments Using Machine Learning-Based Ensemble Approaches

MUHAMMAD EHTISHAM HASSAN [1], IFFAT MAAB [2], MASROOR HUSSAIN [1], USMAN HABIB [3], AND YUTAKA MATSUO [4]

[1]Department of Data Science, Computer Engineering, Ghulam Ishaq Khan Institute, Swabi 23640, Pakistan
[2]Digital Content and Media Sciences Research Division, National Institute of Informatics, Tokyo 101-0003, Japan
[3]Software Engineering Department, FAST School of Computing, National University of Computer & Emerging Sciences, Islamabad 44000, Pakistan
[4]The University of Tokyo, Tokyo 113-8654, Japan

CORRESPONDING AUTHOR: IFFAT MAAB (e-mail: iffatmaab@weblab.t.u-tokyo.ac.jp).

ABSTRACT The complex linguistic characteristics and limited resources present sentiment analysis in Roman Urdu as a unique challenge, necessitating the development of accurate NLP models. In this study, we investigate the performance of prominent ensemble methods on two diverse datasets of UCL and IMDB movie reviews with Roman Urdu and English dialects, respectively. We perform a comparative examination to assess the effectiveness of ensemble techniques including stacking, bagging, random subspace, and boosting, optimized through grid search. The ensemble techniques employ four base learners (Support Vector Machine, Random Forest, Logistic Regression, and Naive Bayes) for sentiment classification. The experiment analysis focuses on different N-gram feature sets (unigrams, bigrams, and trigrams), Chi-square feature selection, and text representation schemes (Bag of Words and TF-IDF). Our empirical findings underscore the superiority of stacking across both datasets, achieving high accuracies and F1-scores: 80.30% and 81.76% on the UCL dataset, and 90.92% and 91.12% on the IMDB datasets, respectively. The proposed approach has significant performance compared to baseline approaches on the relevant tasks and improves the accuracy up to 7% on the UCL dataset.

INDEX TERMS Chi-square, ensemble learning, grid search optimization, low resource language, machine learning, natural language processing (NLP), n-gram, sentiment analysis.

I. INTRODUCTION

Profound rise of social media platforms, ease of connectivity, and unprecedented access to information, users actively engage and express their concerns and feedback about products, services, or support systems on media platforms, creating new opportunities and challenges in the digital age [1], [2]. Examples of such platforms include the World Wide Web, social media sites like Facebook and Twitter, as well as blogs, forums, and entertainment websites. The opinions and reviews shared on these platforms serve as a key resource for discerning customer sentiments [3].

Sentiment analysis (SA) can be considered a key form of non-topical text analysis, with several important application domains including politics, news analytics, and marketing, while also posing complex challenges in artificial intelligence [4]. SA involves contextual mining of text and focuses on textual subjectivity, such as opinions, emotions, and attitudes in the source material [5]. By employing a combination of machine learning classifiers (ML) and NLP techniques, SA seeks to analyze the polarity of sentiments expressed by users in written text [6].

Researchers have adopted various feature engineering techniques and ML algorithms to address diverse challenges and enhance the accuracy of NLP tasks [7]. The adopted techniques include bag of words (BOW), term frequency and inverse document frequency (TF-IDF), word

embeddings, sentiment lexicons, neural networks, and ML classifiers comprising Support Vector Machines (SVM), and Naive Bayes (NB) [8], [9]. Deep neural networks (DNN), including Recurrent neural networks (RNN) and Long Short Term Memory Networks (LSTM), have been employed for various complex text classification tasks such as SA [10], [11], with LSTM addressing issues like the vanishing and exploding gradient problems in RNN [12]. However, employing DNN architecture necessitates extensive hyperparameter tuning and large training parameters, which become challenging, particularly with large datasets.

Extensive research on SA focuses on European languages and English, benefiting from their rich linguistic resources. The digital platforms support English and Roman scripts predominantly [14], [20], encouraging users to use their native language with Roman scripts for reviews like Urdu, Hindi, Arabic, etc [21]. SA in languages like Roman Urdu faces limitations due to resource constraints, including the absence of standardized corpora. Urdu, the native and official language of Pakistan, is spoken in many South Asian states due to its historical and cultural roots [22]. The adoption of Roman Urdu, which transcribes Urdu using the English alphabet, on social media platforms serves users of diverse age groups, enhancing communication comfort. Consequently, the development of robust SA models for low-resource Roman Urdu is vital for comprehending user emotions and sentiments.

In this research, we present a novel approach to sentiment analysis in low-resource Roman Urdu, leveraging machine learning and ensemble learning techniques. We investigate ensemble approaches performance optimized using grid search on both UCL Roman Urdu and IMDB datasets. Our study focuses on improving sentiment classification performance, recognizing the complexity of linguistic characteristics and limited resources in Roman Urdu. The main contributions of our study are as follows:

- Developed a robust approach to SA in low-resource Roman Urdu and English dialects, leveraging optimized ensemble classifiers using grid search to enhance performance.
- Devise a standardization method for word variations in Roman Urdu text.
- Applied Chi-square feature selection to identify and select the statistically significant features in the N-gram sets i.e., unigrams, bigrams, trigrams, and their combination.

The subsequent sections of the article are organized as follows: Section II provides an overview of previous research in the subject domain. Section III discusses the proposed methodology for sentiment polarity classification. The experimental setup adopted in this study is detailed in Sections IV, and V discusses the results acquired from the proposed approach. Finally, Section VI concludes the work, highlighting future directions.

II. LITERATURE REVIEW

Sentiment analysis (SA) has gained importance in recent times due to the proliferation of user-generated text, presenting practical applications across various domains. SA seeks to identify and extract sentiment-related information from data sources, essential for knowledge collection and decision-making [23]. Numerous studies on SA have been conducted that utilize supervised, unsupervised, and semi-supervised techniques, however, there exists limited work on Roman Urdu SA. Table 1 outlines the previous work on Roman Urdu SA. This section summarizes the past research on Roman Urdu SA and discusses effective approaches adopted by practitioners across various domains for sentiment analysis.

Mehmood et al. [13] utilized machine learning (ML) approach (NB, SVM, LR, KNN, and DT) with N-gram features for Roman Urdu SA. They created a Roman Urdu corpus comprising 779 reviews spanning five distinct domains: movie, politics, drama, mobile reviews, and miscellaneous. The study findings show that NB and LR classifiers outperform other ML classifiers employed in the study. In another study on Roman Urdu SA [14], SVM was applied to reviews sourced from an e-commerce website (Daraz.pk). They used a vector space model and a TF-IDF weighting scheme to represent the reviews. Mehmood et al. [15] proposed a deep learning (DL) model to analyze emotions and attitudes expressed in Roman Urdu, utilizing a dataset comprising 10,021 sentences related to various genres. They established a manually annotated benchmark corpus for SA in Roman Urdu and applied rule-based, N-gram, and Recurrent Convolutional Neural Network (RCNN) to classify sentiments. RCNN model acquires high performance compared to rule-based and N-gram, achieving an accuracy of 65.2% for binary classification. Another study by Chandio et al. [16] employed fine-tuned SVM utilizing Roman Urdu stemmer to classify sentiments. The study adopted bag of words (BOW) and TF-IDF schemes and introduced the largest Roman Urdu e-commerce dataset (RUECD).

In [17], three neural embeddings (Word2vec, GloVe, and FastText) were introduced for Roman Urdu to enhance NLP tasks. To establish a performance baseline for Roman Urdu SA, the study employed ML classifiers (SVM, NB, and LR), DL models (RCNN and RNN), and proposed a multi-channel hybrid approach. The hybrid approach combining CNN and RCNN with pre-trained embeddings outperforms ML and DL methods on the Roman Urdu corpus with 3,241 sentiments categorized into positive, negative, and neutral classes. A deep learning approach CNN-LSTM was adopted by Khan et al. [18] with diverse word embeddings for SA in both Roman Urdu and English dialects. They employed machine learning classifiers alongside the proposed DL architecture. They evaluated the performance of the DL and ML classifiers on four datasets, where SVM and Word2Vec continuous bag of words (CBOW) demonstrated improved performance on Roman Urdu SA. Another study by Chandio et al. [19], applies deep recurrent architecture RU-BiLSTM for SA in Roman Urdu, combining bidirectional LSTM with word embeddings and attention mechanism. The proposed DL model outperforms the baseline models on two Roman Urdu datasets i.e., RUSA-19 and RUECD.

TABLE 1. Prior Research Work on Roman Urdu Sentiment Analysis

Work	Algorithm	Accuracy (%)	Dataset	Dataset Publicly Available
Mehmood et al. [13]	Machine learning (NB, SVM, Logistic Regression (LR), K-Nearest Neighbor (KNN), and Decision Tree (DT)	76.98	Roman Urdu (reviews)	Yes
Noor et al. [14]	SVM	60.03	Daraz.pk (E-commerce website)	No
Mehmood et al. [15]	Rule-based, N-gram, RCNN	75.1	RUSA-19	Yes
Chandio et al. [16]	SVM	68	RUECD	Yes
Mehmood et al. [17]	LR, NB, SVM, RCNN, Hybrid Multi-channel Approach	82	3241 Roman Urdu sentiments	Yes
Khan et al. [18]	CNN-LSTM	74.8	RUSA-19	Yes
Chandio et al. [19]	RU-BiLSTM	77.50	RUSA-19	Yes

Xie et al. [24] proposed a maximum entropy model to extract emotion words from Wikipedia and corpus using probabilistic latent semantic analysis. A fuzzy logic-based technique for exhibiting the polarity acquired through training sets or a training set was presented by Dragoni and Petrucci [25]. The approach they adopted makes use of potential conceptual domain overlaps to develop a general model that can determine the polarity of texts into arbitrary domains. A new approach for employing machine learning methods on the movie reviews dataset was proposed by Pang and Lee [26], in which text classification algorithms were employed on the subjective parts of the documents using Minimum cuts formulation to determine the polarity of sentiments. The approach adopted, in the initial step subdivided the objective and subjective words belonging to the documents and allowed the remaining words for the next step. To extract the results, NB and SVM classifiers were applied in the proposed approach.

In article [27], the authors reported designing and building a system for summarising movie reviews and ratings for mobile platforms. The results of applying sentiment classification to movie reviews were used to determine the rating in the proposed approach. Recognizing the significance of identifying product features for feature-based summarization, the authors proposed employing Latent Semantic Analysis (LSA) for identifying these features. Another study by Khan et al. [28], employed rule-based, ML-based, and DL-based approaches for Urdu SA on a multi-class dataset to establish baseline results. They manually annotated the Urdu dataset comprising 9,312 reviews into positive, negative, and neutral classes. Text representation schemes of N-gram, FastText, and BERT word embeddings were adopted in the study. The proposed fine-tuned multilingual BERT outperformed other baseline classifiers used in the research, achieving an F1-score of 81.49%.

A. RESEARCH GAP

Prior research focusing on resource-deprived Roman Urdu language shows a notable absence of extensive application of ensemble approaches and feature selection for sentiment polarity classification. This study seeks to address the gap by presenting a broader approach that employs machine learning and ensemble techniques to effectively classify sentiments in Roman Urdu text.

III. METHODOLOGY

In this study, we investigate the performance of machine learning and ensemble classifiers for binary classification of sentiments in both Roman Urdu and English dialects, utilizing two distinct datasets i.e., UCL and IMDB movie reviews. Sample sentences from the UCL Roman Urdu dataset are presented in Table 2. We employed five ensemble approaches in the experiment analysis, namely bagging, random subspace, stacking, boosting and majority voting to enhance classification performance by combining predictions from multiple base learners. The base learners utilized in our study include SVM, Random Forest (RF), LR, and NB, chosen for their competitive performance on various NLP tasks [17], [29]. Our experimental setup comprises the exploration of various word level N-gram feature sets (unigrams, bigrams, trigrams, and unigram + bigram), along with Chi-square feature selection to identify significant features, and two text representation schemes, BOW and TF-IDF. Hyperparameter optimization is conducted through grid search with 10-fold cross-validation to ensure robust model performance. The entire process of the proposed approach can be completed in various steps as depicted in Fig. 1. These steps are elaborated upon in detail in the subsequent subsections, providing a comprehensive overview of the experimental methodology.

A. PREPROCESSING MODULE

The following preprocessing steps are incorporated in this study:

- *Removal of* sentences with complete English language in UCL dataset.
- *Conversion of* text to lower case.
- *Removal of* digits, special characters, and punctuations.
- *Eliminating* ASCII (American Standard Code for Information Interchange) control characters and HTML (Hypertext Markup Language) tags.
- *Stop-words Removal*
  - Common words occurring in the text, both Roman Urdu and English such as “the”, “is”, “in”, “for”, “to”, “at” etc., which are articles and prepositions that add

TABLE 2. Sample Sentences Roman Urdu UCL Corpus

Reviews (with English translation)	Polarity
"Inho ne har field mein apna loha manwaya hai." (They have proven their mettle in every field)	Positive
"Bht achi product hai ap agr lena chahte hain sochiye mat jaldi se le lain." (It's a very good product. If you want to buy it, don't hesitate, buy it quickly)	Positive
"bhai bakwas drama hy ye." (brother, this drama is rubbish)	Negative
"Bht e ganda fabric aur size bhi ghalt bheja aur colour bhi change thy." (The fabric was very poor quality, the size was wrong, and the color was also different)	Negative

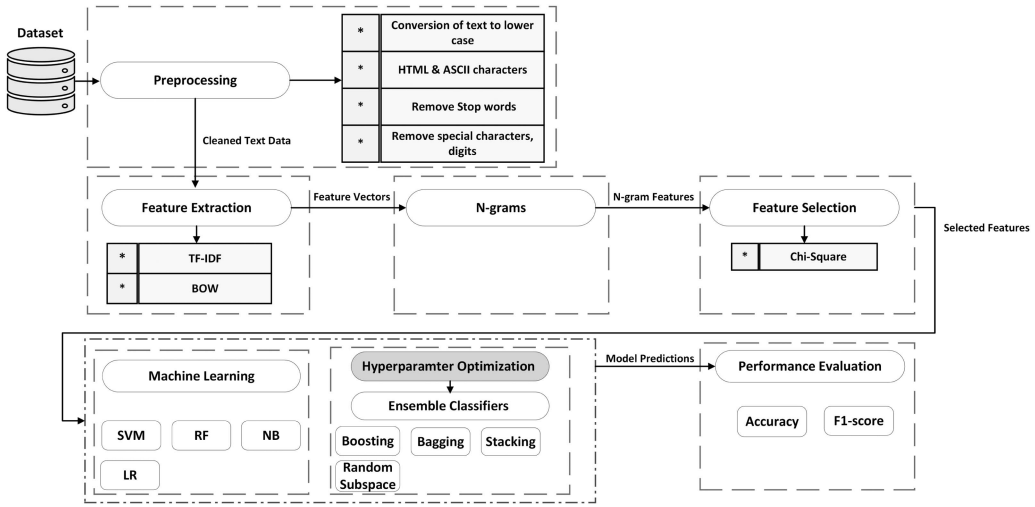


FIGURE 1. Proposed framework for sentiment polarity classification.

no useful meaning to the sentiments were removed from the text. Additionally, we compiled a separate list of stop words specific to Roman Urdu comprising terms like "aur", "ap", "hun", and "hai".

B. STANDARDIZATION OF ROMAN URDU TEXT

Roman Urdu is the Latin script of Urdu. To date as per our knowledge, there is no standardization of Roman Urdu script, resulting in multiple variations of the same word due to differences in spelling. Consider the example of the word "bekaar", which translates to 'useless' in English. The word can appear in various forms such as "bekaaar", or "bekar", all conveying the same negative sentiment but with different spellings. These variations can lead to inconsistency in sentiment analysis (SA) results, as the algorithms employed for SA may not efficiently interpret and identify these variations. To address this challenge we devise a Python program to standardize these variations of words.

Our approach involves creating a mapping dictionary that pairs each standard Roman Urdu word with its variations. The mapping dictionary, comprising 2,500 entries, was manually curated from the vocabulary of the UCL dataset. It primarily groups Roman Urdu words with different variations in spellings conveying similar meanings. Prioritizing word variations as the primary criterion, we also considered phonetic similarity (e.g., "acha" and "accha", both meaning "good" in English) and contextual relevance (e.g., "zabardast" and "bekaar"), ensuring that the standardized forms accurately reflect semantic consistency. The pseudocode of the standardization is given in Algorithm 1. The program systematically replaces each variation in the UCL dataset with its standardized counterpart using a mapping dictionary and regular expressions. For example, variations of the Roman Urdu word "bakwaas" which translates to (rubbish) in English, like "bakwaaas", "bakwaass", and "bakwass" are systematically mapped to "bakwaas". Similarly, variations of "jhoota" (liar) such as "jhota", "jhooty", and "jhooti" are replaced with "jhoota". To standardize the text in the IMDB dataset, lemmatization was performed to reduce words to their root forms using the Natural Language Toolkit [30].

C. FEATURE EXTRACTION

In SA, feature extraction and representation using feature vectors are the key steps. These vectors enable the acquisition of classification models to determine labels for unseen

Algorithm 1: Pseudocode of Standardizing Roman Urdu Word Variations.

Input: Roman Urdu dataset *RU_D*, Word Mapping Dictionary *W_M_D*
Output: Normalized Roman Urdu dataset with standardized words
for each text t in dataset *RU_D* **do**
  for each standard word sw and its variations var in *W_M_D* **do**
    for each variation v in var **do**
      pattern ← create_regex(v)
      t ← replace_with_regex(t, $pattern$, sw)
      end for
    end for
end for
return Normalized Roman Urdu dataset *RU_D*

classes [31]. Our study explores TF-IDF, BOW, and N-gram models for sentiment polarity classification.

1) BAG-OF-WORDS (BOW)

The BOW technique represents a document as a collection of words, without considering grammar, syntax, and word orderings [32]. BOW models the data in the text form by considering the frequency of word occurrence in the corresponding text document. A vector of fixed length is utilized, with each entry mapping to a word present in a pre-defined dictionary. If a word in a sentence appears in the pre-defined dictionary, its corresponding entry in the vector indicates its frequency in the document. However, if the word is not in the dictionary, its entry is typically set to 0. The BOW vocabulary size is determined by the document's word count.

2) TERM FREQUENCY AND INVERSE DOCUMENT FREQUENCY (TF-IDF)

The TF-IDF text representation scheme assigns weights to individual terms within a document based on their term frequency (TF) and inverse document frequency (IDF). TF-IDF emphasizes terms with higher weights as more significant within the document [33]. These two indicators TF and IDF determine the weighted score of a single word in each document using the formula provided below:

$$TF - IDF_{j,k} = TF_{j,k} \times IDF_j \tag{1}$$

where $TF_{j,k}$ determines the text frequency of word w_k in document d_j. IDF_j determines the inverse value of document frequency df_j, and df_j is document frequency of word w_j. Inverse document frequency assigns significance to the words that have rare occurrences in the document.

D. N-GRAM

The N-gram approach is a key text representation scheme extensively utilized in text categorization tasks. In this work, we utilize word-level unigram (uni), bigram (bi), trigram (tri), and unigram + bigram (uni + bi) representations from N-gram models as a feature set. The word order within a sentence's vector representation is captured by the N-gram model. The vector representations are very effective, especially in sentiment analysis where they provide a thorough comprehension of the linguistic complexities found within the text. For example, consider a sentence expressed in Roman Urdu, "Inho ne har field mein apna loha manwaya hai", which in English translates to "They have proven their mettle in every field". Through trigram analysis (N = 3) after stopwords removal, distinct combinations like 'Inho har field', 'har field apna', 'field apna loha', and 'apna loha manwaya', are generated. This N-gram scheme extracts meaningful sequences of N tokens from the text while maintaining the sentence's sequence.

E. FEATURE SELECTION

In this study, we applied Chi-square feature selection to reduce the vast array of textual features into a manageable subset that best captures the underlying patterns of the data. The chi-square test was utilized on word-level N-grams to select the top k features based on the statistical significance. Employing feature selection gives several advantages, including improved interpretability, reduced risk of overfitting, enhanced comprehensibility, and model generalization [34]. The χ^2 statistic is calculated using the following formula:

$$\tilde{\chi}^2(\text{feat}) = \sum_{i=1}^{C} \frac{n_i}{N} \times \chi^2(\text{feat}, \text{class}_i) \tag{2}$$

In (2), $\tilde{\chi}^2(\text{feat})$ represents the chi-square statistic for the feature feat. It is computed by summing over all sentiment classes (C) and multiplying the proportion of occurrences of feature feat within each class ($\frac{n_i}{N}$) by the corresponding chi-square score ($\chi^2(\text{feat}, \text{class}_i)$), which measures the relationship between the feature and the sentiment class.

F. MACHINE LEARNING CLASSIFIERS

This section presents the ML classifiers used in the study to classify Roman Urdu and movie reviews. The learning algorithms take input features which are extracted using techniques outlined in Section III.

1) SUPPORT VECTOR MACHINE

SVM are ML models capable of handling both classification and regression tasks [35]. SVMs are applied to segregate data points by incorporating decision boundaries with hyperplanes into a multidimensional feature space. During training, SVM selects the points near the decision boundary, termed support vectors, which influence the positioning of the hyperplane. In this study, we have applied SVM for the binary classification of Roman Urdu and English reviews into positive and negative classes.

2) LOGISTIC REGRESSION

LR is a linear classification algorithm primarily utilized for binary classification tasks, especially related to NLP [17].

The LR model for binary output variables employs a logistic function to approximate probabilities, providing insights into the relationship between multiple variables [36]. This logistic function, a fundamental component of LR, calculates the probability that a given feature vector k belongs to the positive class using the sigmoid function applied to the linear product of the weight parameters (w) and the input variables (X).

The logistic regression equation is expressed as follows:

$$P(c = 1|d) = g(d) = \frac{1}{1 + e^{wT_d}} \tag{3}$$

where $P(c = 1|d)$ is the probability that document d belongs to class c, and w denotes the feature-weight parameter to be estimated.

3) NAÏVE BAYES CLASSIFIER

NB is a probabilistic ML model with application in text classification tasks [37]. NB classifiers are derived from Bayes' theorem but operate under a firm assumption of feature independence. Multinomial Naïve Bayes model [38] estimates the class probabilities for a given document d, which is represented by the feature vector $(f_1, f_2, \ldots, f_n)$. Under the assumption of independence, the probability of observing features within the range f_1 to f_n for a particular class c can be calculated as the product of individual probabilities, as provided below:

$$P(f_1, f_2, \ldots, f_n|c) = \prod_{1 \le i \le n} P(f_i|c) \tag{4}$$

This formulation simplifies the computation of posterior probabilities when NB is employed to classify new instances. Multinomial Naïves Bayes performs well for easily countable data, such as word counts in text, and permits each feature distribution to be multinomial.

4) RANDOM FOREST

RF is an ML technique that employs numerous numbers of decision trees to achieve more stable and accurate predictions. RF has been widely utilized in NLP tasks based on its significant performance [39]. This approach addresses the inherent limitations of individual decision trees, which often exhibit high variance and low bias, leading to sub-optimal classification performance. RF integrates two key techniques: random feature selection and bagging, both renowned in the realm of machine learning [40].

G. ENSEMBLE TECHNIQUES

Ensemble learning methods utilize a combination of several classifiers to enhance classification accuracy by aggregating their predictions. This approach compensates for individual weaknesses and improves overall performance [41]. In our experimental setup, we employ five ensemble approaches, namely bagging, random subspace, stacking, boosting, and majority voting.

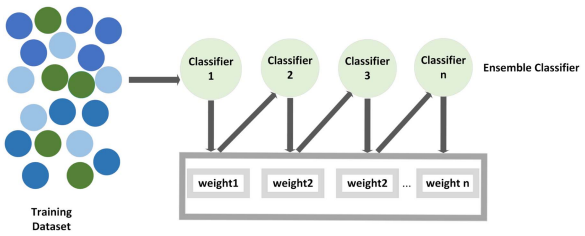


FIGURE 2. Illustration of boosting ensemble technique.

1) BAGGING

Bagging or Bootstrap aggregating [42] is an ensemble learning algorithm of multiple similar base learning algorithms, aimed to improve accuracy and exhibit high predictive performance. The algorithm combines classifiers that are trained using several training sets acquired by applying the bootstrap sampling technique on the initial training set. Bootstrap sampling involves uniform sampling with replacement from the original dataset, maintaining the sizes of the training sets identical to that of the original training set. The base estimators used in our experiment, including SVM, NB, RF, and LR, are trained using this technique.

2) BOOSTING

Boosting [43], is an extensively used ensemble technique employed to improve the performance of weak classifiers and produce a more robust classification model. In this technique, classifiers are trained utilizing various sampling distributions derived from the training data. Weak learning algorithms can be employed to produce a single, more reliable classification model. We employed the Adaboost ensemble technique in our experiment to classify sentiments. AdaBoost enhances the boosting algorithm by emphasizing instances that are challenging to learn. In the initial stage, equal weights are assigned to each pattern within the training set. However, as the ensemble learning progresses, these weights undergo adjustments, increasing for instances misclassified and decreasing for those classified correctly. The boosting ensemble process is presented in Fig. 2.

3) STACKING

Stacking [44] or stacked generalization is an ensemble combination technique that employs a two-staged structure. In this approach, a meta-classifier is trained to combine the predictions of heterogeneous models of different types. The base learners are trained on the complete dataset, whereas the meta-learning algorithm is trained on the output of these base classifiers. To train the meta-classifier, a new dataset comprising the outputs from the base learners is employed. The dataset differs from the one used to train the base learners, aiming to minimize overfitting. In this study, we utilize SVM, NB, LR, and RF as base classifiers in the stacking ensemble approach adopted for sentiment polarity classification. Fig. 3, illustrates the stacking technique adopted in the study.

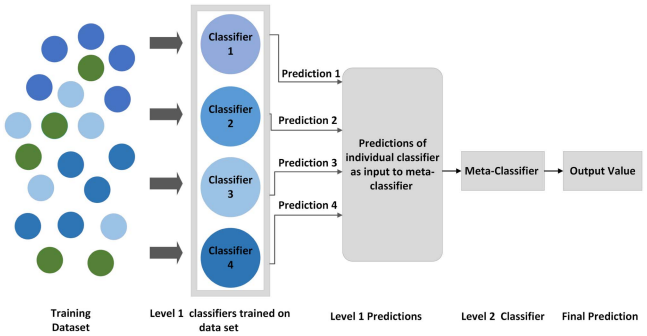


FIGURE 3. Illustration of stacking ensemble technique.

4) RANDOM SUBSPACE

Random subspace method [45], is an ensemble technique that employs feature based partitioning to acquire diversity in the base learners. Multiple base learners are trained on randomly selected feature subspaces and then combined in this ensemble scheme to improve the predictive performance. The weak base classifiers trained on random samples of the feature space are combined to produce a robust classifier similar to bagging. This study employs SVM, LR, NB, and RF as base estimators for random subspace ensemble.

5) VOTING

Voting is another ensemble learning approach in which instances undergo classification by multiple learning algorithms through combined voting. One of the fundamental voting schemes utilized is majority voting, where the output class label is determined by the predictions of more than half of the employed base classifiers. Consequently, the output of the ensemble is based on the class having the most votes. In this study, we employed the majority voting scheme alongside hard voting, a specific instance of majority voting where each classifier in the ensemble contributes equally to the final decision. Hard voting aggregates the predictions made by each classifier and selects the class label with the most votes. This approach offers increased robustness, improved generalization, and reduction in overfitting based on a collective decision-making process.

H. HYPERPARAMETER OPTIMIZATION

The grid search technique is adopted for hyperparameter optimization of ensemble classifiers. The hyperparameter tuning is performed on configurations obtained by combining feature extraction schemes (BOW and TF-IDF), N-gram feature sets, and Chi-square feature selection. Predefined sets of hyperparameters are utilized to conduct the grid searches to identify the optimal values. The systematic approach assists in fine-tuning the models effectively and improving their predictive capabilities on both the UCL and IMDB datasets. Table 3 summarizes the hyperparameters for each ensemble method used in the experiment analysis. Bagging and random subspace `n_estimators` hyperparameter determines the number of base estimators in the ensemble. Three different values are tested for the ensemble i.e., 10, 20, and 50. In the stacking ensemble, the hyperparameter `final_estimator__C` presents the regularization strength of the meta-classifier (Logistic Regression) used for combining the predictions of the base learners. We experimented with three values: 0.1, 1.0, and 10.0. For AdaBoost, the `n_estimators` hyperparameter controls the maximum number of base estimators to be sequentially trained. The AdaBoost ensemble classifier is evaluated on values: 50, 100, and 150.

TABLE 3. Hyperparameters for Ensemble Classifiers

Ensemble Classifier	Hyperparameter	Values
Bagging	`n_estimators`	[10, 20, 50]
Random Subspace	`n_estimators`	[10, 20, 50]
Stacking	`final_estimator__C`	[0.1, 1.0, 10.0]
AdaBoost	`n_estimators`	[50, 100, 150]

TABLE 4. Statistics of UCL and IMDB Datasets

Dataset	Positive	Negative	Neutral	Total
IMDB	25,000	25,000	-	50,000
UCL	6,013	5,286	8,929	20,228

IV. EXPERIMENT

This section outlines the details of the datasets adopted in the study and the procedure for the experiment carried out to classify the sentiments.

A. EXPERIMENTAL DATASET

In this research, we have evaluated the performance of the proposed architecture on two datasets with statistics depicted in Table 4. The experiment was performed on only positive and negative sentiments from both the UCL and IMDB datasets to ensure consistency across both datasets and enhance model performance in binary sentiment classification. Moreover, the adopted approach avoids data imbalance in the UCL dataset with neutral reviews having a significantly different distribution compared to positive and negative reviews. The details of the datasets are presented in the following subsections.

1) UCL ROMAN URDU DATASET

The UCL Roman Urdu corpus comprises 20,228 sentences that are further categorized into three classes: positive, negative, and neutral [15], [46]. The complete dataset comprises 6,013 positive sentences, 5,286 negative sentences, and 8,929 neutral sentences.

2) IMDB DATASET

The 50K IMDB movie reviews is a sentiment analysis dataset used in the experiment analysis comprising English text only and obtained from the Kaggle platform [47]. The dataset is labeled and has 50,000 records with two attributes i.e., review

TABLE 5. Classification Accuracy Results of ML Classifiers on UCL Dataset

Classifier	Unigrams, Bigrams, and Trigrams						Unigram + Bigram	
	BOW			TF-IDF			BOW	TF-IDF
	Uni	Bi	Tri	Uni	Bi	Tri	Uni + Bi	Uni + Bi
Without Feature Selection								
LR	75.39	**63.81**	54.69	**74.91**	63.45	54.73	74.69	**74.29**
SVM	73.14	63.14	54.77	74.29	**63.80**	54.51	72.96	73.80
NB	**75.84**	57.52	54.38	73.62	57.56	54.60	**74.91**	73.36
RF	72.07	61.90	54.77	72.69	63.09	54.82	73.49	73.23
With Chi-square feature selection (top k=1000)								
LR	**76.76**	**65.61**	**55.30**	77.83	**65.26**	55.22	78.53	78.89
SVM	74.76	64.38	55.22	77.06	65.28	55.26	75.79	78.29
NB	76.65	58.62	54.51	**77.95**	58.84	**54.69**	**78.84**	**79.17**
RF	74.51	63.05	55.17	74.11	63.84	55.04	74.60	74.55

TABLE 6. Classification Accuracy Results of ML Classifiers on IMDB Dataset

Classifier	Unigrams, Bigrams, and Trigrams						Unigram + Bigram	
	BOW			TF-IDF			BOW	TF-IDF
	Uni	Bi	Tri	Uni	Bi	Tri	Uni + Bi	Uni + Bi
Without Feature Selection								
LR	86.01	80.43	66.54	87.01	81.44	66.48	**87.03**	87.29
SVM	**86.04**	79.63	65.88	**87.88**	80.54	66.52	86.65	**87.96**
NB	84.95	**81.97**	**67.25**	85.52	**81.91**	**67.17**	85.21	85.96
RF	84.87	77.15	65.39	84.75	76.93	65.49	84.75	84.67
With Chi-square feature selection (top k=4000)								
LR	**88.53**	84.56	69.12	89.01	84.89	71.67	**88.83**	89.29
SVM	88.17	82.33	67.12	**89.10**	85.62	**72.76**	87.68	**89.91**
NB	86.75	**85.88**	**70.64**	87.92	**85.93**	71.92	86.66	87.51
RF	85.43	78.38	67.86	85.21	78.46	70.33	85.52	85.51

and sentiment. The acquired dataset is balanced with 25,000 sentiments having 'positive' polarity and 25,000 having 'negative' polarity.

B. EXPERIMENTAL PROCEDURE

To establish robust models for classifying sentiment polarity, feature extraction schemes of BOW and TF-IDF, utilizing N-grams, are applied to both the training and testing sets. Both UCL and IMDB datasets are split into training and testing sets with an 80:20 ratio, ensuring a representative distribution of classes in both sets. The chi-square test is used to select the most informative and discriminative features across various N-gram configurations to reduce the word-level N-grams in the UCL and IMDB datasets. In this research, a series of experiments were conducted using features ranging from the top 200 to 4000 words based on significant Chi-square test score values for the UCL dataset, and from the top 500 to 6000 words for the IMDB dataset. For optimal results, the top 1000 and 4000 word-level features are selected for the UCL and IMDB datasets, respectively, based on significant Chi-square test score values on the training sets. The difference in the selected features reflects the distinct characteristics of each dataset. The large IMDB dataset has a more diverse vocabulary size compared to the UCL dataset. The tailored approach to feature selection enhances the classification performance capturing the informative and relevant features to train the models leading to optimal accuracy results. The base classifiers are trained on the selected features of the training sets. The ensemble classifiers leverage the insights derived from the trained base classifiers to enhance classification performance and predict sentiment labels on the testing set. The performance metrics of accuracy and the F1 score are adopted to evaluate the classification performance of the ensemble classifiers.

Hyperparameter optimization is performed using grid search with 10-fold cross validation to fine-tune the parameters of each ensemble classifier on the UCL and IMDB datasets. The optimal value of the `n_estimators` hyperparameter, set to 50, gives the best classification results for both bagging and random subspace methods. For the stacking classifier, the optimal value for the `final_estimator__C` hyperparameter is determined to be 0.1. Additionally, the AdaBoost classifier achieves optimal performance with a setting of 150 for the `n_estimators` hyperparameter. The majority voting aggregates the predictions from LR, RF, SVM, and NB classifiers to produce ensemble predictions. All the experiments are conducted within the Google Colaboratory (Colab) notebook environment, leveraging scikit learn packages for machine learning classifiers.

TABLE 7. Classification Accuracy Results of Ensemble Algorithms on UCL Dataset with (a) Unigrams, Bigrams, and trigrams (b) Unigram + Bigram Feature Set

Classifier	(a) Unigrams, Bigrams, and Trigrams						(b) Unigram + Bigram	
	BOW			TF-IDF			BOW	TF-IDF
	Uni	Bi	Tri	Uni	Bi	Tri	Uni + Bi	Uni + Bi
Chi-square top k=1000 features								
Boosting (AdaBoost)	71.19	57.43	55.22	72.16	57.16	55.04	71.28	72.12
RandomSubspace (LR)	77.81	65.75	56.10	78.58	65.48	56.02	**79.11**	79.51
RandomSubspace (SVM)	76.23	64.73	55.97	78.46	65.67	56.01	78.40	80.22
RandomSubspace (NB)	77.47	58.76	54.95	78.65	58.76	54.77	79.03	79.95
RandomSubspace (RF)	74.82	64.73	55.31	75.66	64.29	55.01	75.04	76.10
Bagging (LR)	77.25	65.79	55.56	78.49	65.35	55.35	78.58	79.11
Bagging (SVM)	75.13	64.86	55.89	78.72	65.48	55.91	77.25	79.33
Bagging (NB)	77.74	58.76	54.76	78.82	58.89	54.82	79.07	80.08
Bagging (RF)	74.11	63.53	55.25	74.46	63.89	55.13	74.29	75.04
Voting (Majority Voting)	78.23	64.82	55.77	78.53	65.30	55.93	78.98	79.46
Stacking	**78.47**	**65.81**	**56.31**	**78.91**	**65.70**	**56.15**	79.02	**80.30**

TABLE 8. Classification Accuracy Results of Ensemble Algorithms on IMDB Dataset with (a) Unigrams, Bigrams, and trigrams (b) Unigram + Bigram Feature Set

Classifier	(a) Unigrams, Bigrams, and Trigrams						(b) Unigram + Bigram	
	BOW			TF-IDF			BOW	TF-IDF
	Uni	Bi	Tri	Uni	Bi	Tri	Uni + Bi	Uni + Bi
Chi-square top k=4000 features								
Boosting (AdaBoost)	84.02	72.49	58.15	84.04	72.57	57.91	83.94	84.25
RandomSubspace (LR)	**90.21**	87.01	73.62	89.83	86.32	73.80	**90.68**	89.41
RandomSubspace (SVM)	90.19	86.77	71.24	90.44	86.80	72.73	90.17	90.74
RandomSubspace (NB)	86.75	86.75	73.38	87.65	87.51	73.92	87.33	88.52
RandomSubspace (RF)	85.86	81.64	66.02	85.44	81.04	72.39	85.70	85.70
Bagging (LR)	89.67	86.28	73.26	89.83	85.01	73.74	89.41	89.19
Bagging (SVM)	89.21	85.75	72.51	90.32	86.91	73.80	89.31	90.56
Bagging (NB)	86.67	86.69	73.38	87.84	87.61	73.98	87.25	88.62
Bagging (RF)	85.76	78.80	65.73	85.54	79.24	71.32	85.74	85.68
Voting (Majority Voting)	89.55	85.98	73.05	89.79	86.32	73.92	89.29	89.39
Stacking	89.79	**87.11**	**73.64**	**90.60**	**88.12**	**74.06**	90.11	**90.92**

V. RESULTS AND DISCUSSION

This section presents the results achieved by machine learning classifiers and proposed optimized ensemble classifiers on the sentiment datasets.

A. PERFORMANCE OF THE BASE CLASSIFIERS

Table 5 presents the accuracy values of machine learning classifiers adopted in the study using different N-gram feature sets and weighting schemes (BOW and TF-IDF) on the UCL Roman Urdu dataset. The performance is evaluated without feature selection and using Chi-square on the N-gram features. The unigram features consistently yield the highest accuracy across the classifiers used in the experiment with both BOW and TF-IDF schemes. NB achieves the highest accuracy of 75.84% with the unigram and BOW weighting scheme, without employing feature selection on N-grams. LR achieves an accuracy of 75.39% and SVM acquires an accuracy of 73.14% on unigram and BOW scheme. Similar to the UCL results, comparatively better performance on unigram features is observed on IMDB as depicted in Table 6. The classifiers SVM (87.88%), LR (87.01%), and NB (85.52%) perform better on unigram features with the TF-IDF weighting scheme and no feature selection on N-grams sets. Compared to UCL an increase in the classifiers' performance can be observed with the combination of unigram, and bigram features, where SVM achieves a significant accuracy of 87.96% with unigram and bigram feature sets. RF has low classification performance compared to other base classifiers and achieves a high accuracy of 73.49% on UCL with merged unigram and bigram feature set and BOW scheme.

The performance of the machine learning classifiers improves significantly with Chi-square feature selection on the UCL dataset. LR achieves 76.76% accuracy with unigram and BOW feature extraction scheme after Chi-square feature selection (top k = 1000), compared to 75.39% without feature selection. The performance difference is more pronounced

TABLE 9. Comparison of the Proposed Approach Performance With Closely Related Works

Study	Method Description	Dataset	Language	Accuracy (%)	F1-Score (%)
Khan et al. [28]	Multilingual BERT for multi-class sentiment analysis in Urdu text	UCSA	Urdu	82.50	81.49
Ghorbani et al. [48]	Deep learning (CNN-LSTM-CNN) approach for sentiment classification	IMDB[a]	English	89.02	-
Nafis and Awang [49]	Hybrid feature selection approach using TF-IDF and support vector machine-recursive feature elimination (SVM-RFE)	IMDB[b]	English	86.85	86.96
Chandio et al. [19]	Deep learning (RU-BiLSTM) architecture utilizing word embedding and attention mechanism for sentiment analysis	IMDB[b]	English	88	88
Mehmood et al. [15]	Ruled based, N-gram (character based) and Deep learning (RCNN) approach for Roman Urdu sentiment analysis	UCL	Roman Urdu	53 (Rule-based), 69.8 (4-gram), 73.8 (RCNN)	63.9 (Rule-based), 64.2 (4-gram), 72.3 (RCNN)
Proposed approach	**Optimized ensemble classifiers utilizing Chi-square feature selection on unigram, bigram, and trigram feature set and its combination**	**UCL**	Roman Urdu	**80.30 (Stacking)**	**81.76 (Stacking)**
		IMDB[b]	English	**90.92 (Stacking)**	**91.12 (Stacking)**

* IMDB[a] : 2,000 movie reviews, * IMDB[b] : 50,000 movie reviews.

with the Chi-square feature selection employed on a combination of N-gram features (i.e., unigram + bigram) compared to no feature selection on the same combinations. NB achieves the highest accuracy of 79.17% using Chi-square feature selection on the merge of unigram, and bigram features with the TF-IDF weighting scheme. LR achieves comparatively better performance than SVM with an accuracy of 78.53% on the feature set of merged unigram, and bigram with the BOW scheme and using Chi-square feature selection. Similar improved results of the base classifiers can also be observed in the IMDB corpus on top 4000 N-grams selected using Chi-square. SVM demonstrates high performance on IMDB, achieving an accuracy of 89.91%, utilizing features selected through Chi-square (top k = 4000) on the combined feature set of unigram, and bigram with the TF-IDF text representation scheme.

B. PERFORMANCE OF THE PROPOSED ENSEMBLE CLASSIFIERS

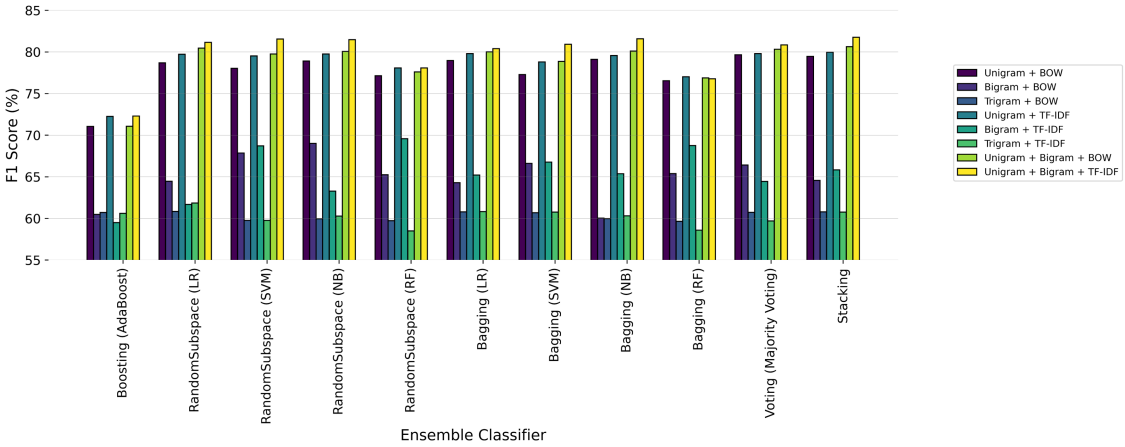
Table 7 presents detailed classification results for the UCL Roman Urdu dataset in terms of accuracy. The results are obtained by incorporating ensemble algorithms which are further optimized through grid search. The stacking ensemble approach gives consistent high performance across multiple combinations on UCL, achieving an impressive accuracy of 80.30%. The performance is attained using the combined unigram and bigram feature set (i.e., unigram + bigram) with TF-IDF representation and Chi-square top k = 1000 features. Notably, stacking leverages all base classifiers, including LR, SVM, NB, and RF, with logistic regression serving as the meta-classifier. Random subspace with SVM as base estimator achieves the second best accuracy of 80.22% on merged unigram and bigram features and TF-IDF weighting scheme. Majority voting, another ensemble technique employing all base classifiers, also demonstrates competitive performance, achieving accuracies ranging from 55.77% to 79.46%. Across both bagging and random subspace methods, LR and NB consistently yield strong results, particularly with TF-IDF representation, showcasing the effectiveness of the feature extraction technique in capturing meaningful textual information. Boosting (AdaBoost) demonstrates comparatively lower performance across the UCL dataset, suggesting limitations in effectively leveraging weak learners to boost overall performance.

The consistent performance of the stacking ensemble is also observed in the IMDB dataset as shown in Table 8, achieving the highest accuracy of 90.92%. This significant performance is achieved with the combination of unigram and bigram features (i.e., unigram + bigram) and TF-IDF representation using Chi-square top k = 4000 features. Similar to the UCL dataset, random subspace with SVM as base estimator achieves a second high accuracy of 90.74% on merged unigram and bigram features with the TF-IDF scheme. Majority voting, which incorporates all base classifiers, exhibits competitive accuracy results ranging from 89.29% to 89.79%. Across bagging and random subspace methods, LR consistently demonstrates strong performance, particularly with TF-IDF representation. Considering the performance of text representation schemes on both datasets, BOWs generally produce less accuracy as compared to TF-IDF. However, both techniques showcase consistent trends across ensemble methods. TF-IDF representation tends to capture the semantic relevance of words more effectively, resulting in higher classification accuracy across various ensemble techniques.

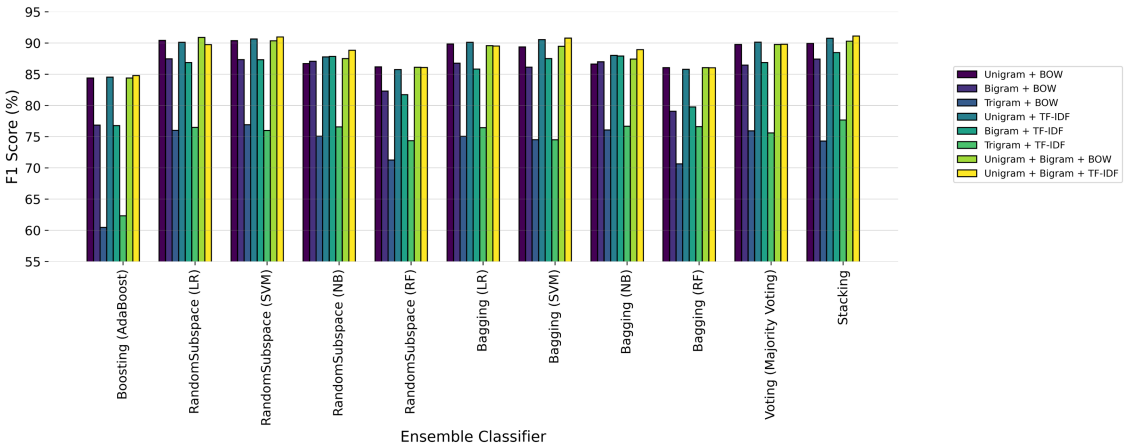
The stacking ensemble method achieves the highest F1-score of 81.76% on the UCL dataset and 91.12% on the IMDB dataset, utilizing the combination of unigram and bigram feature set (i.e., unigram + bigram) with the TF-IDF scheme as depicted in Fig. 4.

C. COMPARATIVE EVALUATION

The comparison of our proposed approach on both the datasets (i.e., Roman Urdu UCL and IMDB) with other baseline algorithms is presented in Table 9. The comparison



(a) F1-Score of ensemble classifiers on UCL dataset.



(b) F1-Score of ensemble classifiers on IMDB dataset.

FIGURE 4. **Performance comparison of the proposed ensemble classifiers across N-gram features.**

analysis shows that our proposed ensemble classifiers especially stacking outperform other existing techniques, achieving a high accuracy and F1-score of 80.30% and 81.76%, respectively, on the UCL dataset. Similarly, our proposed approach also maintains a superior performance on the IMDB dataset, achieving a remarkable accuracy of 90.92% and an impressive F1-score of 91.12%.

VI. CONCLUSION AND FUTURE DIRECTIONS

This research provides valuable insights into the performance of ensemble learning techniques for sentiment analysis focusing on low-resource Roman Urdu UCL and IMDB datasets. A standardization approach was implemented to normalize the Roman Urdu dataset, reducing complexity by standardizing word variations.

Our study demonstrates the effectiveness of the stacking ensemble method in classifying sentiments in both Roman Urdu and English, achieving maximum accuracy of 80.30% and 90.92%, respectively. These empirical findings underscore the robustness and versatility of ensemble techniques in sentiment analysis tasks across diverse datasets and languages. Furthermore, our study highlights ensemble learning's potential in addressing the challenges of sentiment analysis in low-resource languages, offering insights applicable to diverse linguistic domains like sentiment emotion classification, sarcasm detection, and fake news identification.

REFERENCES

[1] A. Lytos, T. Lagkas, P. Sarigiannidis, and K. Bontcheva, "The evolution of argumentation mining: From models to social media and emerging tools," *Inf. Process. Manage.*, vol. 56, no. 6, 2019, Art. no. 102055.

[2] M. Al-Smadi, M. Al-Ayyoub, Y. Jararweh, and O. Qawasmeh, "Enhancing aspect-based sentiment analysis of arabic hotels' reviews using morphological, syntactic and semantic features," *Inf. Process. Manage.*, vol. 56, no. 2, pp. 308–319, 2019.

[3] S. Al-Dabet, S. Tedmori, and A.-S. Mohammad, "Enhancing arabic aspect-based sentiment analysis using deep learning models," *Comput. Speech Lang.*, vol. 69, 2021, Art. no. 101224.

[4] O. Araque, G. Zhu, and C. A. Iglesias, "A semantic similarity-based perspective of affect lexicons for sentiment analysis," *Knowl.-Based Syst.*, vol. 165, pp. 346–359, 2019.

[5] A. Kumar, K. Srinivasan, W.-H. Cheng, and A. Y. Zomaya, "Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data," *Inf. Process. Manage.*, vol. 57, no. 1, 2020, Art. no. 102141.

[6] B. Zhang, X. Xu, X. Li, X. Chen, Y. Ye, and Z. Wang, "Sentiment analysis through critic learning for optimizing convolutional neural networks with rules," *Neurocomputing*, vol. 356, pp. 21–30, 2019.

[7] V. K. Vijayan, K. Bindu, and L. Parameswaran, "A comprehensive study of text classification algorithms," in *Proc. 2017 Int. Conf. Adv. Comput. Commun. Inform.*, 2017, pp. 1109–1113.
[8] X. Wang, J. Wang, Y. Yang, and J. Duan, "Labeled LDA-kernel SVM: A short chinese text supervised classification based on sina weibo," in *Proc. 4th Int. Conf. Inf. Sci. Control Eng.*, 2017, pp. 428–432.
[9] M. S. Haydar, M. Al Helal, and S. A. Hossain, "Sentiment extraction from Bangla text: A character level supervised recurrent neural network approach," in *Proc. 2018 Int. Conf. Comput. Commun. Chem. Mater. Electron. Eng.*, 2018, pp. 1–4.
[10] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4335–4385, 2020.
[11] A. Mahmoud and M. Zrigui, "Deep neural network models for paraphrased text classification in the Arabic language," in *Proc. 24th Int. Conf. Appl. Natural Lang. to Inf. Syst.*, Salford, U.K., Springer, 2019, pp. 3–16.
[12] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2013, pp. 1310–1318.
[13] K. Mehmood, D. Essam, and K. Shafi, "Sentiment analysis system for Roman Urdu," in *Proc. 2018 Comput. Conf. Intell. Comput.*, Springer, 2019, pp. 29–42.
[14] F. Noor, M. Bakhtyar, and J. Baber, "Sentiment analysis in e-commerce using SVM on Roman Urdu text," in *Proc. 2nd Int. Conf. Emerg. Technol. Comput.*, London, U.K., Springer, 2019, pp. 213–222.
[15] Z. Mahmood et al., "Deep sentiments in roman urdu text using recurrent convolutional neural network model," *Inf. Process. Manage.*, vol. 57, no. 4, 2020, Art. no. 102233.
[16] B. Chandio et al., "Sentiment analysis of roman urdu on e-commerce reviews using machine learning," *CMES-Comput. Model. Eng. Sci.*, vol. 131, pp. 1263–1287, 2022.
[17] F. Mehmood, M. U. Ghani, M. A. Ibrahim, R. Shahzadi, W. Mahmood, and M. N. Asim, "A precisely xtreme-multi channel hybrid approach for Roman Urdu sentiment analysis," *IEEE Access*, vol. 8, pp. 192740–192759, 2020.
[18] L. Khan, A. Amjad, K. M. Afaq, and H.-T. Chang, "Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media," *Appl. Sci.*, vol. 12, no. 5, 2022, Art. no. 2694.
[19] B. A. Chandio, A. S. Imran, M. Bakhtyar, S. M. Daudpota, and J. Baber, "Attention-based RU-BiLSTM sentiment analysis model for Roman Urdu," *Appl. Sci.*, vol. 12, no. 7, 2022, Art. no. 3641.
[20] A. J. Dueppen, M. L. Bellon-Harn, N. Radhakrishnan, and V. Manchaiah, "Quality and readability of English-language internet information for voice disorders," *J. Voice*, vol. 33, no. 3, pp. 290–296, 2019.
[21] M. A. Qureshi et al., "Sentiment analysis of reviews in natural language: Roman Urdu as a case study," *IEEE Access*, vol. 10, pp. 24945–24954, 2022.
[22] M. Z. Asghar, A. Sattar, A. Khan, A. Ali, F. Masud Kundi, and S. Ahmad, "Creating sentiment lexicon for sentiment analysis in Urdu: The case of a resource-poor language," *Expert Syst.*, vol. 36, no. 3, 2019, Art. no. e12397.
[23] D. Alessia, F. Ferri, P. Grifoni, and T. Guzzo, "Approaches, tools and applications for sentiment analysis implementation," *Int. J. Comput. Appl.*, vol. 125, no. 3, pp. 26–33, 2015.
[24] X. Xie, S. Ge, F. Hu, M. Xie, and N. Jiang, "An improved algorithm for sentiment analysis based on maximum entropy," *Soft Comput.*, vol. 23, no. 2, pp. 599–611, 2019.
[25] M. Dragoni and G. Petrucci, "A fuzzy-based strategy for multi-domain sentiment analysis," *Int. J. Approx. Reasoning*, vol. 93, pp. 59–73, 2018.
[26] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics*, 2004, pp. 271–278.
[27] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, G.-C. Lu, and E. Jou, "Movie rating and review summarization in mobile environment," *IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.)*, vol. 42, no. 3, pp. 397–407, May 2012.
[28] L. Khan, A. Amjad, N. Ashraf, and H.-T. Chang, "Multi-class sentiment analysis of Urdu text using multilingual BERT," *Sci. Rep.*, vol. 12, no. 1, 2022, Art. no. 5436.
[29] A. Jain and V. Jain, "Efficient framework for sentiment classification using apriori based feature reduction," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 8, no. 31, 2021, Art. no. e3.
[30] E. Loper and S. Bird, "NLTK: The natural language toolkit," in *Proc. ACL Workshop Effective Tools Methodol. Teach. Natural Lang. Process. Comput. Linguistics*, 2002, pp. 63–70.
[31] P. C. Lane, D. Clarke, and P. Hender, "On developing robust models for favourability analysis: Model choice, feature sets and imbalanced data," *Decis. Support Syst.*, vol. 53, no. 4, pp. 712–718, 2012.
[32] G. Hackeling, *Mastering Machine Learning With Scikit-Learn*. Birmingham, U.K.: Packt Publishing Ltd., 2017.
[33] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2758–2765, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417410008626
[34] A. Madasu and S. Elango, "Efficient feature selection techniques for sentiment analysis," *Multimedia Tools Appl.*, vol. 79, pp. 6313–6335, 2020.
[35] U. Ali et al., "Automatic cancerous tissue classification using discrete wavelet transformation and support vector machine," *J. Basic Appl. Sci. Res.*, vol. 6, no. 7, pp. 15–23, 2016.
[36] B. Gaye, D. Zhang, and A. Wulamu, "A tweet sentiment classification approach using a hybrid stacked ensemble technique," *Information*, vol. 12, no. 9, 2021, Art. no. 374. [Online]. Available: https://www.mdpi.com/2078-2489/12/9/374
[37] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between multinomial and Bernoulli Naïve Bayes for text classification," in *Proc. 2019 Int. Conf. Automat. Comput. Technol. Manage.*, 2019, pp. 593–596.
[38] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial Naive Bayes for text categorization revisited," in *Proc. 2004 Adv. Artif. Intell.*, 2004, pp. 488–499.
[39] H. Chen, L. Wu, J. Chen, W. Lu, and J. Ding, "A comparative study of automated legal text classification using random forests and deep learning," *Inf. Process. Manage.*, vol. 59, no. 2, 2022, Art. no. 102798.
[40] P. Jiang, H. Wu, J. Wei, F. Sang, X. Sun, and Z. Lu, "RF-DYMHC: Detecting the yeast meiotic recombination hotspots and coldspots by random forest model using gapped dinucleotide composition features," *Nucleic Acids Res.*, vol. 35, no. suppl_2, pp. W47–W51, 2007.
[41] J. Kazmaier and J. H. van Vuuren, "The power of ensemble learning in sentiment analysis," *Expert Syst. Appl.*, vol. 187, 2022, Art. no. 115819.
[42] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[43] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
[44] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.
[45] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.
[46] Z. Sharf and S. U. Rahman, "Performing natural language processing on roman Urdu datasets," *Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 1, pp. 141–148, 2018.
[47] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2011, pp. 142–150.
[48] M. Ghorbani, M. Bahaghighat, Q. Xin, and F. Özen, "ConvLSTMConv network: A deep learning approach for sentiment analysis in cloud computing," *J. Cloud Comput.*, vol. 9, no. 1, pp. 1–12, 2020.
[49] N. S. M. Nafis and S. Awang, "An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification," *IEEE Access*, vol. 9, pp. 52177–52192, 2021.

MUHAMMAD EHTISHAM HASSAN received the M.S. degree in engineering management from NUST-EME, Rawalpindi, Pakistan, in 2018. He is currently working in the capacity of Doctoral Researcher with Ghulam Ishaq Khan Institute (GIKI), Khyber Pakhtunkhwa, Pakistan, where he is also with the Department of Data Science and a part with Data Engineering Management Analysis (DEMA) Research Group. He is the Coordinator with the International Collegiate Programming Contest (ICPC), Asia Topi Region. His research interests include natural language processing, machine learning, deep learning, large language models (LLMs), and data science.

IFFAT MAAB received a master's degree in computer engineering from the Ghulam Ishaq Khan Institute for Engineering Sciences and Technology, Khyber Pakhtunkhwa, Pakistan in 2016, and the Ph.D. degree in technology management for innovation from the Department of Technology Management for Innovation, Graduate School of Engineering, University of Tokyo, Japan, in 2024. She was a Lecturer with the Ghulam Ishaq Khan Institute for Engineering Sciences and Technology, Khyber Pakhtunkhwa, Pakistan, and also remained a Researcher with Stuttgart University, Stuttgart, Germany. She is currently a Project Researcher with the Digital Content and Media Sciences Research Division, National Institute of Informatics (NII), Tokyo. Her research interests encompass machine learning, Big Data analysis, natural language processing (NLP), computer vision, and deep learning.

MASROOR HUSSAIN is currently a Professor of data science and the HoD with Computer Engineering and Data Science Department, Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute (GIKI). His research interests include adaptive meshing, data mining, data warehousing, finite element methods, neural networks, and parallel computing. He is also the Regional Contest Director (RCD) with International Collegiate Programming Contest (ICPC), Asia Topi Region. He is the Project Director and has expertise in the areas of high-performance computing as well as deep neural networks. He has supervised project(s) in the areas of sentiment analysis and fake news detection which will significantly influence the project. He possesses vast experience in parallel programming and has supervised M.S. and Ph.D. degrees students in this area. He had participated in many national programming and software competitions, as the representative of NUCES-FAST Lahore, like Softec, Procom, and Softcom during his undergraduate studies. He was the recipient of the 1st Prize in SoftCom Nov-2001 for the speed programming competition.

USMAN HABIB received the M.S. degree in telematics: communication network and networked services from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2010, and the Ph.D. degree in engineering sciences from ICT Department, Technical University of Vienna, Vienna, Austria, in 2016. He was with the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Khyber Pakhtunkhwa, Pakistan, and Swabi and COMSATS University Abbottabad Campus. He is currently an Associate Professor and the Head of Software Engineering Department with the FAST School of Computing, National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan. Since 2006, he has more than eighteen years of teaching and research experience, and has successfully completed various industrial projects along with serving in academia. He is also actively engages in research and has authored numerous conference and journal publications. His research interests include machine learning, data analytics, pattern recognition, security, and medical image processing.

YUTAKA MATSUO is currently a Professor with the Department of Technology Management for Innovation, Graduate School of Engineering, University of Tokyo, Tokyo, Japan. He is also the Chairman with Japan Deep Learning Association, an outside Director with SoftBank Group, Chairman with Cabinet Office's AI Strategy Council, and an Expert Member with the Council for the Realization of New Capitalism. His research interests include artificial intelligence, deep learning, and web mining.