

Received 15 September 2023, accepted 12 October 2023, date of publication 16 October 2023, date of current version 15 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3325048

RESEARCH ARTICLE

Cognitive Relationship-Based Approach for Urdu Sarcasm and Sentiment Classification

MUHAMMAD YASEEN KHAN¹, TAFSEER AHMED¹,
MUHAMMAD SHOAB SIDDIQUI², (Member, IEEE), AND SHAUKAT WASI¹

¹Center for Language Computing, Mohammad Ali Jinnah University, Karachi, Sindh 75400, Pakistan

²Faculty of Computer and Information Systems, Islamic University of Madinah, Medina 42351, Saudi Arabia

Corresponding author: Muhammad Yaseen Khan (sp18phcs0001@maju.edu.pk)

This work was supported by the Deanship of Research, Islamic University of Madinah, Medina, Saudi Arabia.

ABSTRACT Humans have a natural tendency to express their emotions, but they are also skilled at using sarcasm to shape their feelings. In cognitive computing and natural language processing research, sentiment analysis and sarcasm detection are typically treated as separate tasks, with each text analyzed in isolation. However, this approach overlooks the connection between sentiment and sarcasm. We believe that sentiment and sarcasm are closely related and should be analyzed together to achieve a better understanding of context and natural language. In this paper, we propose a new framework that leverages the Cognitive Relationship (CR) between sarcasm and sentiment to improve the accuracy of classification. By taking into account the relationship between these two factors, we can achieve better results in sentiment analysis and sarcasm detection. We have also created a new and nearly balanced dataset for sentiment and sarcasm classification in standard Urdu that contains 7,000 tweets, which make up over 210K tokens. To gain a better understanding of the data, we conducted exploratory data analysis on words, hashtags, and emojis. The proposed methodology conducted a variety of classical machine learning classifiers and tested them with different variations of the dataset. After a thorough analysis of the results and errors, we found that the CR-based approach for sarcasm and sentiment classification performed better than the traditional stand-alone (SA) approach. Among the classifiers, Linear Regression and eXtreme Gradient Boosting proved to be the most effective. The sentiment classification based on CR has demonstrated a 9.3% enhancement compared to the stand-alone (SA) method while maintaining an overall improvement of approximately 22% compared to the baseline distribution. In the same way, the sarcasm classification based on CR has shown a 9.1% improvement over the SA approach and approximately 23.6% improvement over the baseline distribution.

INDEX TERMS Sarcasm analysis, sentiment classification, natural language processing, machine learning, cognition, Urdu.

I. INTRODUCTION

Humans, as sentient beings, possess the intrinsic ability to express and share their affective states, regardless of the methods employed to convey such expressions. These affective phenomena have been extensively theorized within the realm of psychology, contributing significantly to our understanding of human emotions and behavior. In recent times, the domain of affective computing, which involves

computational approaches to handling affective data, has garnered significant interest, particularly in social and business sciences. The ability to computationally analyze and interpret affective expressions from various sources, such as text, speech, images, and videos, holds tremendous importance in diverse domains like marketing, customer service, education, healthcare, and entertainment. Automated affective classification has the potential to revolutionize how businesses and organizations interact with customers, tailor experiences, and respond to feedback, ultimately enhancing user satisfaction and engagement. However, alongside the

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu¹.

mounting research work in this field, there is a growing recognition of the ethical implications of automated affective classification. Researchers are increasingly emphasizing the importance of ethical considerations to ensure fairness, transparency, and privacy in the development and deployment of affective computing systems [1].

Affect is a multifarious subject within the literature of psychological studies; therefrom we can see moods, emotions, sarcasm/irony, and sentiment etc., [2], [3], [4], [5], and [6]. have been considered as different concepts and deal with a substantial amount of theories thereon. This paper focuses on the computational classification of two concepts, sarcasm and sentiment, in running text. Sentiment classification is commonly viewed as a binary classification problem, where the text is classified as either *positive* or *negative* [1], [7] (and in some cases, *neutral* [8], [9]). Sarcasm classification, on the other hand, is a strictly binary classification problem, where the primary objective is to predict whether the text is *sarcastic* or *non-sarcastic*. Classifying sarcasm and sentiment in the running text is relatively a complex task as supportive information, such as speech and facial expressions, is missing. Forbye it, on a challenging perspective, we can sense a twist when the sentiment of a given text is ostensibly negative—however, it projects a sarcastic attitude towards the matter. Thus, we can relatively think that similar behaviour has a greater propensity to alter the meaning of the running text when a document carries emojis, emoticons, and puns.

In mainstream research, sarcasm, and sentiment classification are commonly treated as separate problems. We support this claim with a basic rationale, as there scarcely exists any textual dataset where these two problems can coexist. However, we contend that these problems are interrelated due to a natural phenomenon, presenting the potential to mutually enhance comprehension to a greater extent. In this regard, some experiments have observed the development of context-awareness through the usage of distributional semantics. Where the core concept is that words with similar meanings appear in comparable contexts and share common associations. However, this approach tends to focus solely on the textual content and may not fully incorporate the psychological understanding of context when specific labels like sarcasm or sentiment are present. Consequently, the inclusive usage of psychological context understanding remains limited.

To address this limitation, our paper proposes a novel framework that leverages a mutual Cognitive Relationship (CR) between sarcasm and sentiment in text classification. By exploring this cognitive relationship, we aim to facilitate a more comprehensive and inclusive usage of contextual information, leading to an improved understanding of the subject matter. Our proposed framework is based on the following propositions:

Proposition 1: The Cognitive Relationship is the latent potential or intrinsic information that coexists among multiple distinct concepts of affects, such that its uti-

lization certainly enhances the cognition of the matter understudy.

Forbye the condition as it is stated in proposition 1, it also postulates the symmetric property in the Machine Learning (ML) tasks, such that it is given below:

Proposition 2: In a machine learning context, the Cognitive Relationship is meant to exist symmetrically for any combination of affective concept(s) and input data.

For the example of proposition 2, let \mathcal{X} be the given data, affective concepts \mathcal{Y}_1 as sentiment and \mathcal{Y}_2 as sarcasm. The Stand-Alone (SA) manner of ML classification (*viz.*, baseline ML function) is $g_0 : \mathcal{X} \rightarrow \mathcal{Y}$. Let $\Theta(g.)$ be the ML evaluation metric to quantify the performance of any ML function $g.$ For CR-based \mathcal{Y}_1 classification, we utilize \mathcal{Y}_2 in input and modify the ML function as $g_1 : \langle \mathcal{X}, \mathcal{Y}_2 \rangle \rightarrow \mathcal{Y}_1$ and *vice versa* in $g_2 : \langle \mathcal{X}, \mathcal{Y}_1 \rangle \rightarrow \mathcal{Y}_2$. Thus, to prove the validity of the proposed architecture *i.e.*, the existence of CR between \mathcal{Y}_1 and \mathcal{Y}_2 w.r.t \mathcal{X} , it must assert the propositional equivalence that is $\Theta(g_1) > \Theta(g_0) \Leftrightarrow \Theta(g_2) > \Theta(g_0)$.

The input language employed in this paper is Urdu, which is one of the major languages of the world w.r.t the population of its speakers; however, w.r.t the computing research, tools, and applications, it is a victim of resource poverty. Urdu is intelligible with Hindi, which is syntactically and morphologically akin to the Urdu language. Besides the grammar, they share a huge overlap of vocabulary [10], [11], [12], [13]. However, the primary difference between the two is the writing script, such that standard Urdu uses modified Perso-Arabic script, namely, *Nastalique*; whereas, Hindi is written in the *Devanagari* script. Urdu, likewise its share with Hindi vocabulary, has got hundreds of loan words from Arabic, Persian, Turkish, and English [13]. We also observed that people used to speak and write Urdu in a code-mixed and code-switched manner, while using local languages such as Punjabi, Pashto, Gujarati/Memoni, and Bengali. Thus, we can surmise the existence of non-Urdu puns and sarcastic phrases coming in between the running Urdu text and speech. Forbye it, the source of data is Twitter; from where we can get an enormous collection of publicly and openly expressed short texts, engaging almost every topic of discussion. Hence, these things, *i.e.*, a shared vocabulary, puns, free-style open expressions, and usage of emojis (be they accurate w.r.t context or not), make the complexity of sarcasm and sentiment analysis more diverse.

The main contributions and achievements made in this paper are enumerated below.

- 1) With this paper, we turned up to commit the first work for automated sarcasm classification in standard Urdu.
- 2) We presented ‘proportionate sampling’—an easy-to-use method of creating a subset, emphasizing the natural and semantic distribution of the dataset.
- 3) We prepared a preliminary and comprehensive dataset for sarcasm and sentiment analysis, in standard Urdu, enriched with emojis and other relevant elements of online social networks. The majority of the tweets

in the dataset are monolingual and otherwise have a slight flavor of Punjabi mocking style. The number of records in the dataset is 7,000 (comprising over 210,628 tokens, 2,210 hashtags, and 24,572 emojis); moreover, the dataset is almost balanced w.r.t sarcasm and sentiment classes.

- 4) We demonstrated that sarcasm and sentiment classification can be improved by utilizing the CR. The proposed framework appeared to outperform the technique of sarcasm and sentiment classification, which are carried out in a SA manner.
- 5) A lot of research works showed that data preparation/labelling can be done with distant supervision—which claims that emojis are useful for labelling the sentiment/dataset creation. In this regard, we investigated such a hypothesis and found the shortcomings of such a criterion after a thorough analysis thereon.

Alongside it, we performed rigorous testing of the proposed framework with different combinations of supervised ML techniques and variations of the datasets to corroborate the viability and usefulness of the proposed framework over SA systems.

The rest of the paper is organized as the section II shares a detailed analysis of related work, which is exclusively divided into subsections for sentiment and sarcasm classification (see section II-A and section II-B), as well as for the context-aware classification in section II-C. The details of dataset creation (i.e., collection, annotation, pre-processing) are maintained in section III–III-B. In sub-section III-C, we provide exploratory data analysis for the dataset. The section III-D contains details of ML contrivances (such as vectorization, classifiers, experimental setup, and thorough evaluation criterion). In section IV, we presented a comprehensive commentary on the result and errors w.r.t each experiment in the whole setup; followed by a comparative analysis thereof. In last, the conclusion and future work is maintained in section V. The full details of the result at the dataset level with respective confusion matrices are presented in the appendix.

II. BACKGROUND AND RELATED WORK

The review of the state-of-the-art in the automated classification of sentiment and sarcasm in standard Urdu is provided in the subsequent subsections. However, we considered the academic and research papers where the work for the Urdu language in standard script i.e., Nastaliq is maintained. Hence, the works which show a computational technique for sarcasm and sentiment classification in Romanized Urdu or Urdu speech are excluded from the study.

Forbye it, going through the literature available online and alongside the relevant academic research portals, we found that there is very little work done so far for the automated sarcasm classification in South Asian regional languages such as Bengali, Hindi, Punjabi etc. In the same context, for the sarcasm classification in Urdu text, as per the date so far, we were unsuccessful in finding any of such work.

Hence, for the literature review, we considered Hindi as the closest language. However, work in the other languages w.r.t significance is also reported in related work.

A. SENTIMENT CLASSIFICATION

Based on the translation of five different English sentiment lexicons, which are Affective norms of English words [14], SenticNet [15], AFINN [16], and NRC-EmoLex [17], Khan et al. [18] showed the sentiment classification of Urdu tweets. In this regard, they collected the dataset [19] of 1K Urdu tweets, which are related to politics, sports, religion and society. The accuracy and F1-score achieved through the lexicon-based experiment vary between 55-60% and 58-70% respectively. Moreover, the same work also shows the comparison of the aforesaid approach with ML-based classification, where the linear Logistic Regression [20], [21] with bi-gram features secured the highest F1-score i.e. 70%.

Mukhtar et al. [22] showed SA in Urdu blogs by employing a rule-based technique with an emphasis maintained on the usage of intensifiers. In this regard, a sentiment corpus of 151 Urdu blogs is created which is consisting of 6,025 sentences. The sentiment analyzer achieves an accuracy of 83.4%, and this improves by 5.09% with the use of intensifiers. Another study by Mukhtar et al. [23] investigated the impact of negations in the USA, using the same dataset as their earlier work on identification [22]. However, this time, with an overall accuracy of 78.32%, the improvement is seen around 4.4% when the negations are handled during the analysis.

Hassan and Shoaib [24] worked out an extended technique on baseline Bag-of-Words (BoW) to capture the sub-opinion in the running text. In this regard, they build up a sentence segmentation module that is based on the determination of coordinating and subordinating conjunctions in the sentence, a sentiment lexicon, and their utilization for determining the sentiment orientation of the sentence. The dataset of a total of 844 product reviews about cars, cosmetics, and electronic devices is used for testing the proposed approach. The proposed/extended BoW approach showed an improvement of 8.91% accuracy (with an overall accuracy of 75.98%) in comparison to the baseline BoW approach.

Bibi et al. [25] did a very cursory work for SA in Urdu tweets—which are reportedly related to the news items. The paper briefly defines the data collection strategy through which 600 tweets were collected; out of which 500 tweets are used in training the predictive system. The methodology only entails the employment of a decision tree classifier [26]. The reported result shows 90% accuracy, which can be contentious as the dataset employed in the experiment is too small and the provided technique is not compared with anyone.

In their study, Nasim and Ghani [8] utilized Markov chains to classify sentiment in both binary (positive and negative) and ternary (positive, negative, and neutral) forms. The dataset used in the study contained 3,103 tweets. The paper

also showed the experiments with the lexicon and ML-based approaches and in a comparative analysis, it is found the proposed method outperformed lexicon and ML-based techniques by securing the highest accuracy i.e., 69% and 86.5% for ternary and binary sentiment classification respectively.

Sehar et al. [27] presented a DNN-based multi-modal approach for the USA. In this regard, 44 review videos were accumulated from YouTube through which 1,372 utterances are segmented. The experimental setup entails a variety of different deep learning approaches (such as Convolutional Neural Network (CNN) [28], Long Short-Term Memory (LSTM) [29], and the bi-directional variations thereof etc.) and combination of textual, audio, and video data for comparative analysis. The result showed 95.3% accuracy achieved when all of the three mediums of data (audio, video, and text) are exploited in the input; whereas the accuracy of the prediction with the individual inputs was 84%, 89%, and 80% for textual, video, and audio respectively.

Naqvi et al. [30] showed a DNN-based approach for the USA. The dataset used in the paper is about the news 6K articles/reports that are collected from online sources. However, the labelling strategy and Inter-Annotator Agreement (IAA) of the corpus of such length are found absent in the paper. The methodology used in the paper involves CNN, Bi-LSTM, attention-based LSTM [31], and the combination thereof. The experimental combinations conclude attention-based LSTM to be the most optimal model for the USA by achieving 77.9% accuracy. In the same context, Safder et al. [9] also showed a DNN-based approach for the USA. The dataset employed in the paper was collected from online sources and consisted of feeds relating to sports, food, software, politics, and entertainment; the total number of accumulated records in the dataset is 6,281 and 10,008 for the binary and ternary classes respectively. However, a similar suite of algorithms was applied for the classification and comparative analysis thereof. The result concluded CNN+RNN to be the most optimal model through securing 84.98% and 68.56% accuracy for binary and ternary sentiment classification. Employing DNN-based CNN and LSTM architectures, Khan et al. [32] also showed an experiment for the USA; for which a dataset of 9,601 user reviews is used. Besides DNN, the methodology includes classical ML algorithms (such as Support Vector Machines [33], AdaBoost [34], [35], and linear Logistic Regression (LR)-based classifiers) and utilization of FastText-based word embeddings [36] for document vectorization etc. However, the result reports the highest F1-score, i.e., 82.05% is achieved with n -gram features using LR. In another experiment, Khan et al. [37] showed usage of multilingual BERT for the USA, which secured 81.49% F1-score. We would also like to maintain that unlike Naqvi et al. [30], Safder et al. [9] and Khan et al. [32], [37] showed a proper strategy for data collection alongside computing the IAA on the labelled dataset.

Khan and Junejo [38] conducted an experiment utilizing a hybrid classification approach, which incorporated lexicon-based scores of positive and negative classes for supervised learning. The results showed a notable improvement of 7.88% and 1.7% for the hybridized approach compared to the lexicon-based approach and machine learning-based approach, respectively.

B. SARCASM CLASSIFICATION

In the same context, Suhaimin et al. [39] proposed extraction of sarcastic features in a bilingual context for Malay and English languages. In this regard, Malay text is translated into English, and linguistic features are extracted, which alongside lexical and syntactic features involve pragmatics and prosodic features as well. The result shows that the combination of prosodic, pragmatic, and syntactic features with a non-linear SVM is the most optimal technique—securing 85.2% F1-score—for classification.

Mukherjee and Bala [40] showed the application of naïve Bayes and fuzzy clustering for sarcasm classification. The study involved n -gram and Part of Speech (PoS)-based features extraction, and a dataset based on micro-blogs. The paper reports the achievement of 65% accuracy.

Bharti et al. [41] targeted on context-based approach for Hindi news. However, no standard classifying algorithm is reported. In another work, Bharti et al. [42] showed a keyword-matching technique to classify sarcastic and non-sarcastic tweets; where the reported accuracy is 79.4% on the dataset of 500 tweets.

Swami et al. [43] showed sarcasm detection in code-mixed data relating to the English and Hindi tweet. In this regard, a dataset of 5,250 tweets is prepared and a suite of algorithms (i.e., SVM (with both radial basis function and linear kernels), RF) is used in a 10-fold cross-validation strategy. The result showed 78.4% accuracy with RF when word and character n -grams, emoticons, and sarcastic lexicons are exploited.

Samonte et al. [44] focused on sentence-level sarcasm detection based on lexical clues and using ML algorithms i.e., Maximum Entropy (ME), naïve Bayes and SVM in Filipino/Tagalog and English tweets. In this regard, they gathered 6K tweets for English and 6K tweets for Filipino, which are related to multiple real-life domains. The experimental setup also includes working and comparisons with ML tools, namely, WEKA and RapidMiner. Using RapidMiner, the highest accuracy, i.e., 98.7% for the Filipino validation set and 93.6% for the English validation set with SVM and RF respectively.

Hazarika et al. [45] adopted a DNN-based hybrid approach in which a combination of two techniques i.e., context-based and content-based are employed. In this regard, word embeddings, stylometric, and personality-based features are used in CNN-based architecture. The proposed method secured 77% and 86% F1-score respectively on the balanced and imbalanced datasets, which brought 7% and 5% improvement in comparison to the then state-of-the-art [46]. In a similar

context, Ren et al. [47] showed DNN-based sarcasm detection in tweets; where they use the history of the tweet author for getting the context of the tweets. On the dataset of basic 1,500 tweets and 6,774 historic trails of tweets, the proposed CNN-based architecture secured a 63% F1-score.

Thimmappa, in his thesis [48], showed word embedding-based sarcasm classification where the paragraphs are vectorized using word2vec [49]. The experiment was executed on an English news dataset i.e., consisting of 26,709 records. Once the embedding is generated, the methodology employs SVM, LR, and RF for classification. The SVM outperformed the rest of the classifiers by securing 91.2%.

Cai et al. [50] worked DNN-based multi-modal approach, which includes images and videos. The experiment involves the usage of different variations of CNN and Bi-LSTM-based architecture. The dataset was comprising over 19,816 items on which the proposed method achieves 80.1% F1-score. In a similar context, Bedi et al. [51] multi-modal approach for Hindi and English in a code-mixed environment. The dataset is consisting of 3,139 sarcastic records in 1,100 dialogues—carrying 14K utterances. The experiment proposed LSTM and attention-based architecture and was carried out with different input combinations. However, the highest F1-score, i.e., 71.1%, is attained when acoustic and textual input is given and an utterance-level attention mechanism is employed in combination with LSTM and Context-based attention. Jain et al. [52] showed the usage of soft-attention, CNN, and Bi-LSTM in a mash-up language (*viz.* a code-mixed environment of Hindi and English languages). The dataset is consisting of 3K sarcastic and 3K non-sarcastic tweets; the proposed approach attained an 89% F1-score.

C. CONTEXT-AWARE SARCASM AND SENTIMENT CLASSIFICATION

For the context-awareness, we found the basic motive lies within the story of the tweet. For example, [41] maintained that sarcasm happens when the story of the tweet contradicts the sentiment of the tweet; Muhammad et al. [53] and Bhat et al. [54] maintained almost similar stance for the context-awareness. We maintain that these sorts of hypotheses, in a larger perspective, make the essence of the work limited to the context of idealizing and data labelling strategies.

Forbye it, we also found the context of the running text is retained by employing the dense vector representations that are based on DNN-inspired word embeddings. In this regard, we have seen many papers, which claim context-awareness by using pre-trained models and transformers, such as Potamias et al. [55] used RoBERTa [56], Alharbi and Lee [57] used BERT that is fine-tuned for the Arabic language, namely, ArBERT [58] for dealing with the context in the running text. Similarly, Kumar and Sarin [59] used FastText [36] and BERT [60] embeddings for context-based dense representation. Badlani et al. [61] showed the concatenation of embeddings that are based on hate speech, humour, sarcasm, and sentiment for sentiment classification. The

datasets used in this work are based on Yelp reviews [62], [63], [64] in the English language.

D. RESEARCH GAP

Based on our analysis, the majority of existing research has approached the problems of sentiment and sarcasm classification in isolation, with limited instances of their combined utilization for classification tasks. Previous attempts at combining affective information have primarily focused on employing distributional semantics or transformer-based dense vectors to replace traditional vector-transformation and feature selection methods. However, we have observed a lack of comprehensive work integrating both sarcasm and sentiment information within a single framework, especially for the resource-poor Urdu language.

While a study conducted by Badlani et al. [61] demonstrated a related approach for English language datasets, its applicability to Urdu language remains limited due to the scarcity of resources. Consequently, the proposed CR-based classification technique presents a novel and significant contribution to the field of sarcasm and sentiment classification in the standard Urdu language. By leveraging a mutual Cognitive Relationship (CR) between sarcasm and sentiment, our approach aims to overcome the limitations of existing methods and provide an innovative solution for effectively handling affective data in Urdu text classification tasks.

III. MATERIALS AND METHODS

This section is divided into two main subsections, which are respectively focusing on the data alongside the associated tasks therein *viz.* data collection, labelling, and pre-processing; and the details of feature extraction techniques, classifiers, and metrics employed in the experiment.

A. DATA: COLLECTION AND LABELING

Since there is no benchmark or public dataset available for the experiment, therefore we carried out this phase in a very attentive manner.

1) DATA COLLECTION

We used a Python-based online library, namely **Tweepy**,¹ for scraping tweets. Alongside different configuring parameters of data gathering, we set the language to ‘Urdu’ and the result type to ‘popular’, and as for the search query, we iterate over the list of trending hashtags that widely reflect different aspects of society and human life. The list also includes popular personalities, which have a significant following in politics, sports, finance, technology, and show business among other institutions such as defence and judiciary. As a result, we collected over 1 million distinct tweets that do not include retweets, replies, or redundant content. We maintain that for the removal of redundancy, we simply used (hash) sets as the data structure to store the text of the tweet. Followed by the data gathering, we move forward with creating a small

¹<https://www.tweepy.org/>

subset for data labelling. However, this selection is based on cluster analysis. This would technically mean performing k -means clustering on the dataset and then, from each cluster, selecting tweets as per the cluster's proportion w.r.t to the whole corpus. To mean the aforementioned sampling method mathematically, consider the algorithm 1. Thus, for a hundred clusters (i.e., $k = 100$), we iterate 500 cycles for hypothetically reaching the cluster maturity. Eventually, we drew around 15K randomly selected tweets from the corpus of 1 million tweets per the criterion discussed in algorithm 1.

Algorithm 1 Proportionate Sampling

```

Require: Set of distinct tweets ( $\Gamma$ ), set of clusters ( $C$ )
formed on  $\Gamma$ .
1:  $n \leftarrow |\Gamma|$ ;  $\triangleright$  be the number of tweets in  $\Gamma$ .
2:  $k \leftarrow |C|$ ;  $\triangleright$  be the number of clusters in  $C$ .
3:  $S \leftarrow \emptyset$ ;  $\triangleright$  be an empty set for the selected tweets.
4:  $s \leftarrow 15000$ ;  $\triangleright$  be the number of tweets to be selected.
5: for  $k' \in \{1 \dots k\}$  do
6:    $m \leftarrow \left\lfloor \frac{|C_{k'}|}{n} \times s \right\rfloor$ ;  $\triangleright$  Where  $|C_{k'}|$  be the number
of tweets in cluster  $C_{k'}$ .
7:   for  $i \in \{1 \dots m\}$  do
8:      $t \leftarrow$  be the randomly selected tweet from  $C_{k'}$ .
9:      $S \leftarrow S \cup \{t\}$ 
10:     $C_{k'} \leftarrow C_{k'} \setminus \{t\}$ 
11:   end for
12: end for
13: return  $S$ 
    
```

2) DATA LABELING

This phase is the crux of dataset creation; however, always seen as a tedious task. One of the key challenges of such

labelling is the confusion among the annotators on the matter under observation. Hence, the role of clear and handy data annotating guidelines is of utmost importance. In this regard, we worked out such guidelines for labelling tweets per their sentiment and sarcasm. The action items for guidelines are enumerated subsequently.

- Language.** Mark 'UR' if the language of the tweet is standard Urdu; otherwise 'NUR'. Although we have set the language to Urdu while scraping tweets; however, we seldom find tweets written in other languages. The basic problem lies with the script of standard Urdu, i.e., Nastalique—which allows text composition in Arabic, Kashmiri, Persian/Farsi, Punjabi, Sindhi, and even the Uyghur languages. Similarly, we can expect content in a code-mixed and code-switched manner, or comprising only emojis/emoticons, hashtags, and mentions. Thus, for such tweets, where a dominating portion of the content is not reflecting or relate to Urdu vocabulary, we requested annotators to mark them 'NUR'. Finally, the tweets labelled per the language 'UR' are considered for sentiment and sarcasm annotations.
- Sentiment.** In the case of a subjective tweet: mark '1' if the synergy and central idea of the tweet are positive, that is, there is no sign of apathy, cynicism, insult, negativity, pejorative comment, abusive or swear wordings and offensive attitude; and on the existence of any such matter in the text, the case is negative and marking must be maintained as '-1'. Otherwise, if there is no subjectivity, mark '0' for being a neutral tweet. However, in the case where you have got the minute propensity of polarity shown in the tweet (and more of the propensity towards neutrality), we asked annotators to emphasize choosing subjective labels.

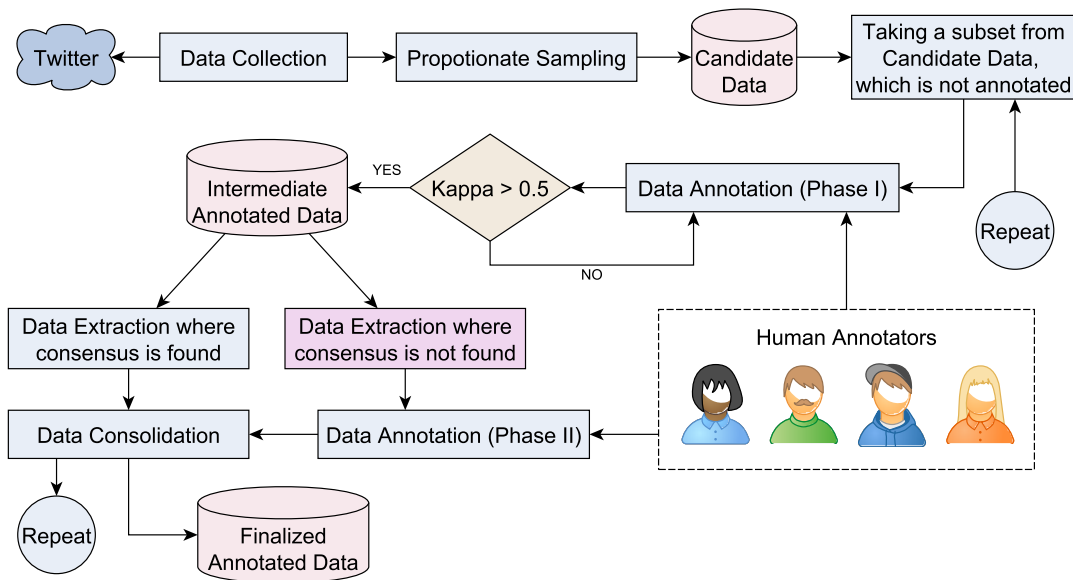


FIGURE 1. The overall data annotation activity.

TABLE 1. List of alphabets that require to be replaced for the character normalization. Source alphabets have to be searched out in Urdu text and the target alphabet is to replace them.

Source Alphabets	Target Alphabet
آ ا ا ا ا ا ا ا	ا (U+0627)
ب	ب (U+0628)
س س س س س س س س	ك (U+0643)
و و و و و و و و	و (U+0648)
ه ه ه ه ه ه ه ه	ه (U+0648)
ھ	ھ (U+06BE)
ی ی ی ی ی ی ی ی	ي (U+06CC)
ے ے ے ے ے ے ے ے	ع (U+06D2)

TABLE 2. Distribution of finalized dataset w.r.t sentiment and sarcasm classes, at the end of annotation phase II.

		Sentiment		Total
		Negative (-1)	Positive (1)	
Sarcasm	Non-Sarcastic (0)	665	2,431	3,096 (44.2%)
	Sarcastic (1)	3,022	882	3,904 (55.7%)
Total		3,687 (52.67%)	3,313 (47.32%)	7,000

- Sarcasm.** This is the sort of Boolean label, indicating the existence/reflection of sarcasm in the content or not. The situation of sarcasm occurrence relies on positive and negative concepts if they are co-existing in a tweet against a mutual target. Forbye it, if there is a mocking, taunt, or funny comment that is close to offending someone/something implies the matter is sarcastic; hence, the very tweet should be marked '1', otherwise '0'.

Figure 1 depicts the overall data annotation activity, where with a group of seven (human) annotators, we devised a data labelling strategy such that the dataset is divided into multiple subsets. In phase I: a subset is assigned to the pair of random annotators (in a blind manner) where they provide labels for the sentiment and sarcasm on the tweet under observation. After the labelling activity is completed, we conducted a comparative analysis and measure Cohen’s kappa statistics (κ) for the subset; in case of lower agreement ($\kappa < 0.5$), we repeat the labelling activity for the very subset. In the alternate case (i.e., $\kappa > 0.5$), we extract the tweets in the labelled pool for which the labels are in agreement. However, for the tweets where the consensus was not developed, we kept them aside (for phase II) to be labelled by the third annotator, followed by adding the very tweet in the labelled pool as per the majority votes. The breakdown of the annotated information for sarcasm and sentiment is given respectively in table 3 and 4. However, *nota bene* that the reported numbers in tables 3 and 4 are the cumulative statistics of the repetitive annotation process as defined for phase I; thus, we can presume κ statistics to be low at the very phase but with phase II, as it is described above, we gradually improved the decisions and IAA. We maintain that the κ -statistics of the labelled tweets are found between $\approx .76-.77$ respectively for sarcasm and sentiment—indicating

TABLE 3. Cumulative labelling distribution (in %) between annotators for sarcasm, at the end of annotation phase I.

		Annotator 1	
		Non-sarcastic (0)	Sarcastic (1)
Annotator 2	Non-sarcastic (0)	37.12	2.6
	Sarcastic (1)	9.28	50.98

TABLE 4. Cumulative labelling distribution (in %) between annotators for sentiment, at the end of annotation phase I.

		Annotator 1	
		Negative (-1)	Positive (1)
Annotator 2	Negative (-1)	44.34	1.63
	Positive (1)	9.94	44.06

overall a substantial inter-annotator agreement [65], [66]; and thus, as of *a posteriori* effect at the end of phase II, with the consent of the third annotator, the decisions reached to the perfect agreement, or the tweet is rejected for its inclusion in the final dataset (as it reflects a cursory or no subjective attitude in the text).

As the result of the data annotation activity, in total, 7K tweets were finalized after labelling, out of which, 3,904 are sarcastic and 3,096 are non-sarcastic. Similarly, for the sentiment classification problem, 3,687 tweets are negative, and 3,313 tweets are positive. The lowest count of sentiment-sarcasm pairs is found in around 665 tweets, which are negative and non-sarcastic in nature. In contrast, the highest count of sentiment-sarcasm pairs is found around 3,022 negatively sarcastic tweets. We maintain that the dataset is near balanced w.r.t to sentiment classification (i.e., in the ratio of $\sim 53:47$ in %); and similarly, for the sarcasm classification, it appears to be slightly imbalanced (i.e., in the


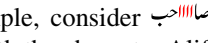
ratio of $\sim 44:55$ in %). These numbers are summarized and maintained in table 2.

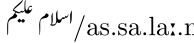
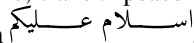
B. DATA PRE-PROCESSING

There are two sorts of pre-processing: the first one deals with the cleaning of the dataset to make it usable for ML purposes, and the latter one is for the construction of datasets featuring slight variation w.r.t the content of tweets.

1) DATA CLEANING

This includes procedures for anomaly removal that occur generally in natural language processing, specifically due to the lexicographic errors committed by the public on online social platforms while writing text in the Urdu language. In the subsequent list, these procedures are discussed in detail.

- 1) We applied a shallow pre-processing technique for dealing with the word segmentation problem in Urdu text [67]. However, since the experiment conducted in course of this paper is the baseline work, therefore, the word segmentation is only limited to the insertion of white space after ے ‘Badi Ye’ and ں ‘Noon-ghunna’ [18], [19]. Similarly, we coarsely inserted a white space before and after the glyphs/symbolic contractions for the religious honorifics (e.g.  etc.)
- 2) Character deformation is also a critical problem in Urdu text, for example, the usage of ڪ (Sindhi) ‘Swash Kaaf’ (U+06AA) and ڪھ (Sindhi Khey) (U+06A9) in place of Urdu alphabet ک ‘Urdu Kaaf’ (U+0643). Another example is ے ‘Farsi Ye’ (U+06CC), ى ‘Arabic Alif Maksura’ (U+0649), and ے ‘Arabic Ye’ (U+064A), all of these characters may visually appear similar; however, they are distinct w.r.t their Unicode values (see the respective values in parenthesis) and behave differently in the text processing. Similarly, there exist multiple alphabetic deformation/variations where the diacritic is the intrinsic part of the alphabet (see table 1). Hence, we customize a lookup table, that is table 1, and normalize the deformed alphabets (source alphabets) accordingly with the correct ones (target alphabet). We maintain that usage of the Python-based library **UrduHack**² is handy for fixing such problems; however, we experienced some shortcomings. In last, the words/hashtags containing non-Urdu alphabets were converted to lowercase.
- 3) We identify the character repetition in the running text and remove it by keeping the leading character. However, we took care of the condition that the number of repeated characters is more than two in a contiguous sequence. For example, consider  where the reduplication is done with the character Alif

(see the sub-string coloured in red); thus, the eventual word after the removal of repeated characters would be صاحب /sa:hib/ (noun, honorific as ‘mister’). This also caters to the matter of character stretching/elongation (conventionally known as Kashida or Tatweel in Perso-Arabic typography) [18]; for example, compare  /as.sa.la.mu ʔa.laj.kum/ (a greeting phrase in Arabic; transl. ‘peace be upon you’) vs. its elongated form .

- 4) Forbye it, we also conducted a similar procedure for replacing emoji and punctuation marks. Since we know that, there are many cases where a certain sequence of punctuation marks can produce a specific emoticon (see https://en.wikipedia.org/wiki/List_of_emoticons for a detailed list). Hence, for all such conditions, we compiled a list of universal emoticons, and utilized it in a lookup function for examining the existence of emoticons in the text; in the negative evidence of their existence, the punctuation that does not form any emotion was removed from the text.
- 5) For better tokenization, we coarsely inserted white space before and after every hashtag, emoji, and emoticon.
- 6) We removed all URLs, mentions, and specific words reflecting anything specific within the domain of Twitter and social networks for example RT (i.e., retweet).
- 7) We removed all of the Arabic diacritics in the text.

C. EXPLORATORY DATA ANALYSIS

After pre-processing the tweets in an aforementioned manner, we computed the statistical information on the finalized dataset. These insights, relating to the correlation and lexical analysis are separately maintained in the subsequent sections.

1) CORRELATION AND ASSOCIATION

Sarcasm and sentiment are the categorical and dichotomous variables, to determine the correlations and association between them we employed 4 different statistics which are enumerated below with their details. These statistics need data in the format of contingency matrix [68], for which, consider the values presented in 2×2 contingency matrix (*cf.* table 2); let the number of non-sarcastic and negative tweets as n_1 , non-sarcastic as and positive tweets as n_2 , sarcastic and negative tweets as n_3 , sarcastic and positive tweets as n_4 , and N be the total number of tweets. For all statistics, the null hypothesis (H_0) states that the variables do not correlate; and on contrary, the alternate hypothesis (H_a) affirms the correlation between them. Table 5 shows the values for these statistics.

- 1) **Tetrachoric correlation** quantifies whether the two variables are correlated or not [69], [70]. The value of the statistic ranges in $[-1,1]$, where the value if

²<https://github.com/urduhack>

TABLE 5. Correlation and association statistics.

Statistic	Value
Tetrachoric correlation	-0.7698
Pearson’s χ^2 test	2164.36
Matthew’s correlation coefficient	-0.5563
Cramér’s V	0.5563

approaching -1 , indicates a strong negative correlation, and similarly, the value if approaching 1 , indicates a strong positive correlation; however, if the value is close to 0 , then there is no correlation between the variables. The Tetrachoric correlation can be calculated as [70]:

$$\text{Tetrachoric correlation} = \cos\left(\frac{\pi}{\left(1 + \sqrt{\frac{n_1 \cdot n_4}{n_2 \cdot n_3}}\right)}\right) \quad (1)$$

Since the value of the Tetrachoric correlation is ≈ -0.77 , therefore, we reject H_0 and conclude that sarcasm and sentiment are likely to have a negative correlation.

2) **Pearson’s χ^2 test** is a statistical assessment for how likely there can be the difference between observed and expected sets, arose by chance [71]. We can calculate the expected value (E_i) respectively for every cell per the following function [71]:

$$E_i = \frac{\sum r_i \sum c_i}{N}; \quad i \in \{n_1, n_2, n_3, n_4\} \quad (2)$$

where r and c are the row and column of the contingency table, corresponding to the cell i . The

χ^2 statistics, thus, can be calculated through the following function [71]:

$$\chi^2 = \sum_{i=1}^4 \frac{(n_i - E_i)^2}{E_i} \quad (3)$$

With the degree of freedom = 1 , and significance factor (α) = 0.01 , we get the p -value = 0 ; since p -value < α , therefore, we reject the H_0 and conclude the two variables are dependent.

3) **Matthew’s correlation coefficient** or ϕ coefficient shows association between two dichotomous variables [72], [73]. Likewise, the Tetrachoric correlation ranges in $[-1, 1]$ and holds a similar intuition for the value. The statistic can be calculated through the following equation [73], [74]:

$$\text{Matthew’s corr. coef.} = \phi = \frac{n_1 n_4 - n_2 n_3}{\sqrt{\sum r_1 \sum r_2 \sum c_1 \sum c_2}} \quad (4)$$

Since the value of ϕ is negative and has got a propensity towards -1 , therefore, we reject the H_0 and conclude that the sarcasm and sentiment are somewhat associated.

4) **Cramér’s V** is another measure for association analysis [75]; however, in the case of 2×2 contingency matrix, it is equal to the absolute value of ϕ . Thus, for the case of this paper, it will be ranging in $[0, 1]$ [75], [76]. Cramér’s V can be calculated through the following equation [75]:

$$\text{Cramér’s V} = \sqrt{\frac{\chi^2}{N \cdot (q - 1)}} \quad (5)$$

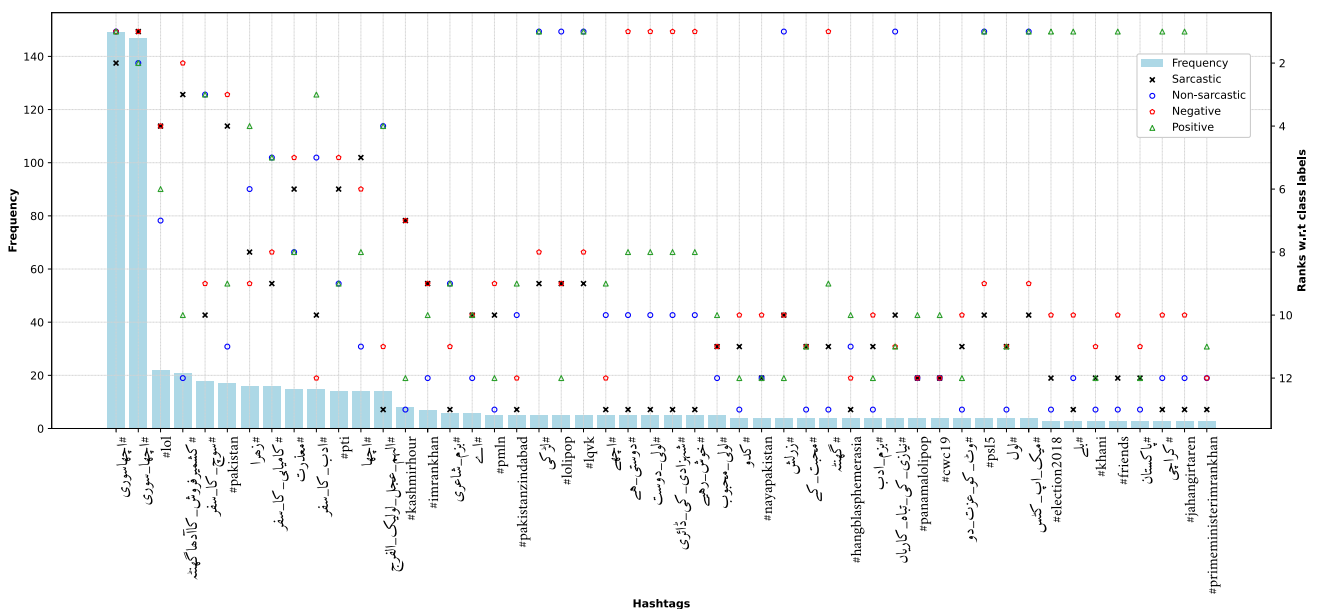


FIGURE 2. Top-50 most frequent hashtags in the corpus. The markers of the scatter plot indicate the respective ranks of hashtags within the associated class domain. See the respective rank of the hashtag at the right y-axis.

where q is the minimum number of rows and a number of columns. Since the value is ditto to the φ , therefore, with the value (≈ 0.56) we reject the H_0 and conclude that there is a lightly moderate association between sarcasm and sentiment.

Observing all of the above measures for correlations and association, we deduce that the two variables i.e., sarcasm and sentiment demonstrate a dependency on each other. However, w.r.t to the proposition 2, there is a need to perform a comparative analysis on the supervised learning results considering the combination of data and affective concepts for utilizing CR, such discussion is maintained in section IV.

2) LEXICAL ANALYSIS

The token-level informatics of the dataset is maintained in tables 6 and 7 for the total number of tokens and (distinct) terms respectively. However, at this moment, we counted everything (words, hashtags, and emojis) as a token. We can see the corpus is consisting of 210,628 tokens; however, if these tokens are seen in their distinct counts (in a non-overlapping manner) then the tokens will be reduced to 18,693 in numbers, and that is a mere $\approx 0.87\%$ of the total count. In addition to it, we found the shared vocabulary between the sentiment classes is 5,859 which is $\approx 31.34\%$ of the total distinct terms of the corpus. Similarly, the shared vocabulary between the sarcasm classes is 5,809, which is $\approx 31.07\%$ of the total distinct terms therein.

Table 8 presents the count and the respective distribution of hashtags among the classification labels. The number of distinct hashtags found in the corpus is 2,210. We maintain there is no such significance, w.r.t numbers, reflected

through the distribution; except for the fact that tweets that are—in a paired manner of negative sentiment and sarcastic context—turned up to have the maximum hashtags. However, in the same paired manner the tweets, which are negative and non-sarcastic in nature, hold the lowest share (i.e., 216) of having hashtags. Beyond the statistics provided in table 8, we found the count of common hashtags between sarcasm classes is 124, and the count of common hashtags between sentiment classes is 100.

Figure 2 tells us the top-50 frequent hashtags used in the corpus irrespective of any of the class labels; alongside it, the rank of these hashtags stands w.r.t the class. We can sense some of the hashtags can play a very discriminative role in classification, for example, #psl15 (a hashtag relating to Pakistan Super League, a sport/cricket tournament) stood first in positive and a non-sarcastic class of tweet whereas it is very low in rank for the opposite counterparts. In a similar context, hashtags such as #ادب_کا_سفر/adab ka sə.fər/ (transl. ‘the journey of literature’) shows more discriminative role for sentiment and sarcasm classes; and #کشمیر_فروش_کا_ادھاگھنٹہ/kəʃmɪr furo:ʃ ka a:q̄h̄a: 'ḡh̄əŋt̄a/ (transl. ‘Half an hour of Kashmir-seller’) appears at the top rank w.r.t the negative and sarcastic tweets vs. positive and non-sarcastic subsets. However, alongside this, many hashtags have got the confounding characteristic as their ranks are high in both classes; for example, #اچھا_سوری/əʃh̄a:ʃd̄i:ʃ/ (transl. ‘okay sorry’) is at the first place for both classes w.r.t sentiment, and similarly, #نیازی_کی_تباہ_کاریاں/nɪ:ʔzi: ki t̄əb̄a:h̄:k̄ariā:/ (transl. ‘destruction of/made by Niazi’) appears at the same rank w.r.t sentiment classes. Thus, we can anticipate the usage

TABLE 6. Distribution of tokens in finalized dataset w.r.t sentiment and sarcasm classes.

		Sentiment		Total
		Negative (-1)	Positive (1)	
Sarcasm	Non-Sarcastic (0)	19,974	71,359	91,333 (43.3%)
	Sarcastic (1)	95,135	24,160	119,295 (56.6%)
Total		115,109 (54.6%)	95,519 (45.3%)	210,628

TABLE 7. Distribution of terms (distinct tokens) in finalized dataset w.r.t sentiment and sarcasm classes. However, the numbers of terms presented in the table are calculated in a non-overlapping manner; hence, it can be possible that their summation $\neq 1$.

		Sentiment		Total
		Negative (-1)	Positive (1)	
Sarcasm	Non-Sarcastic (0)	4,557	9,378	11,104 (59.4%)
	Sarcastic (1)	11,864	4,756	13,398 (71.7%)
Total		13,299 (71.1%)	11,253 (60.2%)	18,693

TABLE 8. Distribution of hashtags in finalized dataset w.r.t sentiment and sarcasm classes.

		Sentiment		Total
		Negative (-1)	Positive (1)	
Sarcasm	Non-Sarcastic (0)	216	659	875 (39.6%)
	Sarcastic (1)	1,067	628	1,335 (60.4%)
Total		1,283 (58.1%)	927 (41.9%)	2,210

TABLE 9. Distribution of emojis in finalized dataset w.r.t sentiment and sarcasm classes.

		Sentiment		Total
		Negative (-1)	Positive (1)	
Sarcasm	Non-Sarcastic (0)	1,985	8,120	10,105 (41.1%)
	Sarcastic (1)	10,920	3,547	14,467 (58.9%)
Total		12,905 (52.5%)	11,667 (47.5%)	24,572

TABLE 10. Distribution of tweets containing emojis in finalized dataset w.r.t sentiment and sarcasm classes.

		Sentiment		Total
		Negative (-1)	Positive (1)	
Sarcasm	Non-Sarcastic (0)	626	2,348	2,974 (35.8%)
	Sarcastic (1)	2,949	860	3,809 (64.2%)
Total		3,575 (41.3%)	3,208 (58.7%)	6,783

of hashtag-based feature extraction would be challenging in probabilistic supervised learning methods.

The corpus contains a comprehensive variety of emojis. In relation to it, table 9 presents the distribution of (total appearances of) emojis w.r.t to the classification labels. The overall usage of emojis is seen more liable in sarcastic tweets (i.e., 14,467, which makes approximately 58.87% of whole emojis count as in the corpus). The tweets which are negative and sarcastic hold the largest share of emojis, i.e. 10,920, in a paired manner. Whereas, the tweets which are positive and non-sarcastic have got a similar significance by comprising over 8,120 records. In the same context, table 10 presents the number of tweets containing emojis w.r.t the classification labels. Forbye it, these emojis share $\approx 12\%$ of the corpus—if compared with the number of tokens. In last, we present a list of top-50 emojis and the distribution thereof w.r.t the sarcasm and sentiment classes table 11. In addition to it, in the situation where the emojis are likely to be spammed or placed in a recurring manner, we maintain that the emojis are counted once while computing the numbers for table 11.

We can surmise that relying solely on the usage/meaning of emojis for tweet collection (in the manner of distant supervision) can be erroneous. For example, in a general sense, the symbol 😄 (joy, 1st entry in table 11), 😊 (beaming face with smiling eyes, 4th entry *ibid.*), 🤪 (rolling on the floor, 5th entry *ibid.*), 😁 (smile, 19th entry *ibid.*), 🤩 (grimacing, 30th entry *ibid.*) are interpreted in positive terms; however, we found their distributions unassertive for the positive class as their share has got only 38–41% of tweets for the positive sentiments out of the total tweets where the respective emojis exist. Similarly, the emojis that are, in a general sense, taken as a negative such as 😐 (neutral face, 20th entry *ibid.*), 😞 (pensive, 29th entry *ibid.*), and 😐 (expressionless, 33rd entry *ibid.*); however, they have got a dominant share, i.e., 52–56%, in the positive class. In contrast, we also have the obvious relations with emojis and the sentiments, such as 🙏 (pray, 2nd entry *ibid.*), 🥰 (heart eyes, 7th entry *ibid.*), 😊 (blush, 10th entry *ibid.*), 🌹 (rose, 32nd entry *ibid.*), ❤️ (two hearts, 47th entry

ibid.) are taken as positive emojis and do have a dominant share in positive class; similarly, 😏 (pout, 38th entry *ibid.*) and 💔 (broken heart, 49th entry *ibid.*) are generally considered negative and have got the dominant negative share as well.

D. METHODOLOGY: FEATURE EXTRACTION, CLASSIFIERS, EXPERIMENTAL SETUP, AND EVALUATIONS

1) DOCUMENT VECTORIZATION AND FEATURE SELECTION

For representing documents in a vector space, we primarily performed two forms of document vectorization, namely *count vectorization* (or frequency-based vectorization), and *Term Frequency-Inverse Document Frequency* (TF-IDF) based vectorization. The details of these vectorization techniques are given in the subsequent paragraphs.

In count vectorization, a $\mathcal{M}_{|\mathcal{D}|\times|\mathcal{T}|}$ matrix is created for representing text documents in the vector space (where \mathcal{D} is the set of documents, $|\mathcal{D}|$ is the number of documents, and $|\mathcal{T}|$ is the number of distinct terms); such that every row in the matrix represents a single document and the column of the matrix represents a specific term. The value for each element of \mathcal{M} say $m_{(i,j)}$; $m_{(i,j)} \in \mathcal{M}$ holds the frequency of the term j in the document i . However, due to the limited capacity of memory, usually, the matrix \mathcal{M} is converted into a sparse representation where we keep only the information of terms having at least 1 count in the document. Figure 3 depicts how a count vectorizer works in a non-sparse fashion.

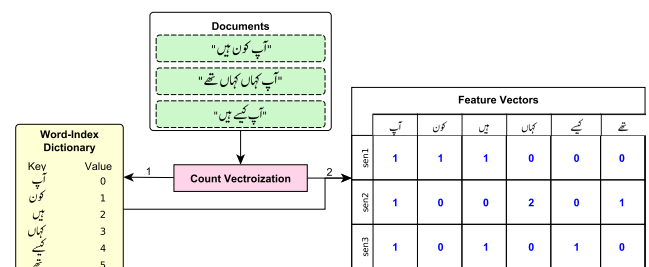
**FIGURE 3.** Example of count vectorization.

TABLE 11. Distribution of top-50 emojis in tweets. The text beside the emoji figure is maintained to define the actual concept. Numbers in the parenthesis are the respective percentages w.r.t to the total count of emojis.

№	Emoji	Count	Sarcastic		Non-Sarcastic		% Share	
			Positive	Negative	Positive	Negative	Pos	Sar
1.	😊 (joy)	1,231	241 (.196)	687 (.558)	223 (.181)	80 (.065)	38	75
2.	🙏 (pray/folded hands)	575	54 (.094)	188 (.327)	275 (.478)	58 (.101)	57	42
3.	😜 (winking face with tongue)	462	76 (.164)	271 (.587)	81 (.175)	34 (.074)	34	75
4.	😄 (grinning face with smiling eyes)	340	56 (.168)	178 (.524)	84 (.247)	22 (.065)	41	69
5.	🤪 (rolling on the floor)	323	61 (.1889)	194 (.6006)	52 (.161)	16 (.0495)	35	79
6.	😏 (squinting face with tongue)	227	52 (.2291)	136 (.5991)	27 (.1189)	12 (.0529)	35	83
7.	😍 (heart eyes)	214	36 (.1682)	52 (.243)	121 (.5654)	5 (.0234)	73	41
8.	🍭 (lollipop)	214	13 (.0607)	173 (.8084)	20 (.0935)	8 (.0374)	15	87
9.	😜 (stuck out tongue)	188	25 (.133)	108 (.5745)	46 (.2447)	9 (.0479)	38	71
10.	😊 (blushing with smiling eyes)	170	27 (.1588)	51 (.3)	84 (.4941)	8 (.0471)	65	46
11.	🕶 (sunglasses)	170	19 (.1118)	92 (.5412)	45 (.2647)	14 (.0824)	38	65
12.	😜 (winking face)	164	27 (.1646)	94 (.5732)	32 (.1951)	11 (.0671)	36	74
13.	😄 (grinning face)	156	33 (.2115)	81 (.5192)	32 (.2051)	10 (.0641)	42	73
14.	😄 (laughing/grinning squinting face)	152	33 (.2171)	84 (.5526)	28 (.1842)	7 (.0461)	40	77
15.	😭 (sobbing/loudly crying)	147	14 (.0952)	55 (.3741)	60 (.4082)	18 (.1224)	50	47
16.	🤔 (thinking)	146	18 (.1233)	87 (.5959)	25 (.1712)	16 (.1096)	29	72
17.	😋 (savouring delicious food)	145	27 (.1862)	70 (.4828)	39 (.269)	9 (.0621)	46	67
18.	🙄 (rolling eyes)	144	17 (.1181)	66 (.4583)	49 (.3403)	12 (.0833)	46	58
19.	😊 (smiling face)	143	30 (.2098)	78 (.5455)	25 (.1748)	10 (.0699)	38	76
20.	😐 (neutral face)	138	22 (.1594)	51 (.3696)	50 (.3623)	15 (.1087)	52	53
21.	😷 (mask)	135	12 (.0889)	69 (.5111)	37 (.2741)	17 (.1259)	36	60
22.	😞 (unamused face)	134	20 (.1493)	53 (.3955)	47 (.3507)	14 (.1045)	50	54
23.	🙅 (see no evil)	134	22 (.1642)	67 (.500)	37 (.2761)	8 (.0597)	44	66
24.	😭 (cry)	133	13 (.0977)	57 (.4286)	45 (.3383)	18 (.1353)	44	53
25.	🙅 (speak no evil)	130	25 (.1923)	63 (.4846)	34 (.2615)	8 (.0615)	45	68
26.	😏 (smirk)	130	15 (.1154)	86 (.6615)	23 (.1769)	6 (.0462)	29	78
27.	😬 (zipper mouth face)	122	11 (.0902)	67 (.5492)	31 (.2541)	13 (.1066)	34	64
28.	🏃 (running)	120	24 (.200)	51 (.425)	34 (.2833)	11 (.0917)	48	62
29.	😐 (pensive)	119	6 (.0504)	28 (.2353)	61 (.5126)	24 (.2017)	56	29
30.	😬 (grimacing)	117	15 (.1282)	65 (.5556)	30 (.2564)	7 (.0598)	38	68
31.	😜 (face with open mouth)	116	21 (.181)	62 (.5345)	28 (.2414)	5 (.0431)	42	72
32.	🌹 (rose)	116	11 (.0948)	5 (.0431)	93 (.8017)	7 (.0603)	90	14
33.	😐 (expressionless)	111	17 (.1532)	40 (.3604)	44 (.3964)	10 (.0901)	55	51
34.	👍 (thumbs-up)	110	11 (.100)	27 (.2455)	63 (.5727)	9 (.0818)	67	35
35.	😄 (sweat smile)	107	18 (.1682)	61 (.5701)	24 (.2243)	4 (.0374)	39	74
36.	♂ (male sign)	101	10 (.099)	49 (.4851)	31 (.3069)	11 (.1089)	41	58
37.	👇 (point down)	99	12 (.1212)	46 (.4646)	34 (.3434)	7 (.0707)	46	59
38.	😡 (pout)	98	1 (.0102)	68 (.6939)	7 (.0714)	22 (.2245)	8	70
39.	👏 (clap)	95	14 (.1474)	35 (.3684)	43 (.4526)	3 (.0316)	60	52
40.	😈 (smiling imp)	92	7 (.0761)	59 (.6413)	2 (.0217)	24 (.2609)	10	72
41.	😜 (zany face)	90	13 (.1444)	52 (.5778)	19 (.2111)	6 (.0667)	36	72
42.	😱 (scream)	89	14 (.1573)	41 (.4607)	26 (.2921)	8 (.0899)	45	62
43.	♀ (female sign)	87	20 (.2299)	29 (.3333)	32 (.3678)	6 (.069)	60	56
44.	😎 (nerd face)	84	12 (.1429)	48 (.5714)	16 (.1905)	8 (.0952)	33	71
45.	👋 (wave)	82	8 (.0976)	46 (.561)	15 (.1829)	13 (.1585)	28	66
46.	🙌 (raised hands)	81	10 (.1235)	22 (.2716)	40 (.4938)	9 (.1111)	62	40
47.	❤️ (two hearts)	81	6 (.0741)	5 (.0617)	67 (.8272)	3 (.037)	90	14
48.	🙇 (person bowing)	80	7 (.0875)	27 (.3375)	40 (.500)	6 (.0750)	59	42
49.	💔 (broken heart)	79	2 (.0253)	26 (.3291)	26 (.3291)	25 (.3165)	35	35
50.	😬 (disappoint but relieved)	78	5 (.0641)	29 (.3718)	31 (.3974)	13 (.1667)	46	44

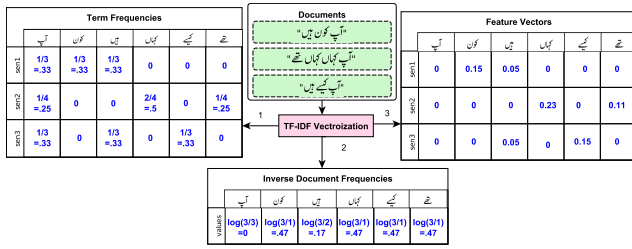


FIGURE 4. Example of TF-IDF vectorization.

Consider the situations in the running text when a term is more frequent (such as pronouns and auxiliaries) in the corpus in comparison to the proper nouns. These frequent words are considered less discriminative (as they tend to appear in every document). Thus, in practice, we are more interested in getting a trade-off between high-frequent and infrequent terms and for such objective TF-IDF-based normalization comes into action. It mitigates the value or the weight of frequent terms and *vice versa* increases the weights for the infrequent terms in the corpus. The TF-IDF is computed by multiplying TF (i.e., term-frequency *viz.* frequency of term t w.r.t document d) with IDF (i.e., inverse documents frequency *viz.* ratio of total documents to the number of documents having term t). To mean it mathematically, consider equation 6 for computing TF, and equation 7 for IDF below. Figure 4 depicts an example of computing TF-IDF.

$$TF(t, d) = \frac{\text{frequency of term } t \text{ in doc } d}{\text{total number of terms in the doc}} = \frac{f_{t,d}}{\sum_{t \in d} f_{t,d}} \quad (6)$$

$$IDF(t, \mathcal{D}) = \log \left(\frac{\text{total number of docs in the dataset}}{\text{number of docs containing the term } t} \right) = \log \left(\frac{|\mathcal{D}|}{|\{d \in \mathcal{D} : t \in d\}|} \right) \quad (7)$$

Forbye the vectorization of documents, we cater for the word features only in the form of bag-of-words and specifically with the two variations of thereof, i.e., *uni-gram*—where the words in a document are kept as of a single entity, and *n-gram*—where the sequence of every n words are maintained in an adjacent manner. For the latter type of word sequencing, we used the combination of sequences in uni-gram, bigram, and trigrams; which equally means to have the value n as $1 \leq n \leq 3$. We maintain that as of the baseline approach, there are no other pre-processing activities such as stop word removal, lemmatization and stemming performed in this work.

2) CLASSIFIERS

We have employed the following seven classifiers in experiments. The information is briefly maintained in the subsequent items.

a: LOGISTIC REGRESSION (LR)

It is amongst the linear model in ML classifiers, for predicting the probability of a target variable [20], [21]. It is well-suited for classification problems that are binary in nature. In its simplest form, it takes the output of the linear regression (see $\theta^T x$ in equation 8) to another function (typically logistic or sigmoid function; see $s(\cdot)$ in equation 9) that converts it into the dichotomous value in the range $[0, 1]$.

$$h_{\theta}(\mathcal{X}) = s\left(\theta^T \mathcal{X}\right) \quad \text{where } 0 \leq h_{\theta} \leq 1 \quad (8)$$

where θ is the weights for the features and s is the logistic or sigmoid function:

$$s(z) = \frac{1}{1 + e^{-z}} \quad \text{provided } z = \theta^T \mathcal{X} \quad (9)$$

Thus, to characterize the target class we set the threshold for the value of s , such that if $s(\cdot) \geq 0.5$ we consider positive class as a predictive outcome for the document \mathcal{X} , otherwise it is negative class [18], [77].

b: SUPPORT VECTOR CLASSIFIER (SVC)

One of the most commonly used ML classifiers is the Support Vector Machine (SVM). It is often considered a suitable classifier for datasets that are imbalanced and have a small or medium size. Its functioning involves fitting a hyperplane in an n -dimensional vector space, such that documents belonging to different classes are separated by the maximum possible margin. Figure 5 illustrates the hyperplane and the separation of classes in SVC. To basic mathematical representation of SCV is given below; where, likewise equation 8, θ represents weights associated with the features:

$$\hat{y} = \begin{cases} 1 & \text{if } \theta^T \cdot \mathcal{X} - b \geq 1 \\ -1 & \text{if } \theta^T \cdot \mathcal{X} - b < 1 \end{cases} \quad (10)$$

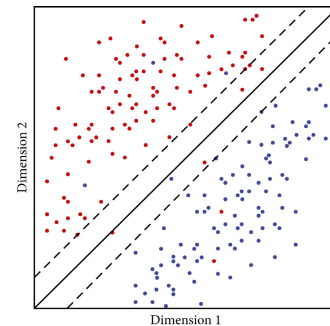


FIGURE 5. Illustration of linear support vector classifier.

c: NAÏVE BAYES (NB)

In text classification, NB is considered amongst the classical methods. It is a probabilistic classifier that uses Bayes' theorem with the assumption that the features are mutually independent [78], [79], [80]. For the document \mathcal{X} , the

prediction for class \mathcal{L}_l can be calculated through the following equation:

$$f(\mathcal{L}_l | \mathcal{X}) = \frac{p(\mathcal{X} | \mathcal{L}_l) p(\mathcal{L}_l)}{p(\mathcal{X})} \quad (11)$$

where the Bayes' rule can (in equation 11) can be expanded w.r.t individual features of the document \mathcal{X} (such that $\mathcal{X} \equiv \{x_0, x_1, \dots, x_n\}$) as:

$$\begin{aligned} f(\mathcal{L}_l | x_0, \dots, x_n) &= p(\mathcal{L}_l) \cdot p(x_0 | \mathcal{L}_l) \cdot p(x_1 | \mathcal{L}_l) \cdot \\ &\quad \dots \cdot p(x_n | \mathcal{L}_l) \\ &\approx p(\mathcal{L}_l) \prod_{i=0}^n p(x_i | \mathcal{L}_l) \end{aligned} \quad (12)$$

However, if the features are in the continuous form (that in our case is with the TF-IDF vectorization) then the Gaussian distribution will be employed [81], [82], [83], which modifies the function as per following:

$$p(x_i | \mathcal{L}_l) = \frac{1}{\sqrt{2\pi\sigma_{\mathcal{L}_l}^2}} \exp\left(-\frac{(x_i - \mu_{\mathcal{L}_l})^2}{2\sigma_{\mathcal{L}_l}^2}\right) \quad (13)$$

We have to repeat the probability calculation for every class; followed by elicitation of class k as the final class (\hat{y}) for a document \mathcal{X} where the probability is maximum *viz.*,

$$\hat{y} = \operatorname{argmax}_{l \in \mathcal{L}} p(l) \prod_{i=1}^n p(x_i | l). \quad (14)$$

d: EXTREME GRADIENT BOOSTING (XGB)

Likewise RFT, it is also a DT-based ensemble approach for classification. However, the basic difference XGB shows against the RFT is: that it takes each DT one by one for learning, considering the improvement through the predecessor's error and giving more weight to the DT that performs better [84]. It also minimizes the loss function using a gradient descent algorithm. In contemporary times, except the deep neural networks, it is a widely preferred ML technique for classification problems.

e: RANDOM FOREST

The ensemble technique in ML, which combines several Decision Trees (DT), is known as Random Forest (RF) [33], [85]. The RF is built by training each tree in combination with randomly selected documents and predicting the class for unseen data. The final class for the unseen data is determined by majority voting. In our experiment, we have used 300 estimators for the DT and *gini impurity* [86] as the function for measuring impurity. Figure 6 depicts the process of RF and the finalization of prediction there out.

f: PASSIVE AGGRESSIVE-CLASSIFIER (PA)

It is a regressive classifier that belongs to the category of online learning in ML [87]. Typically, in PA a model learns incrementally in the form of mini-batches of the model. In function, it is similar to the Perceptron as it does not

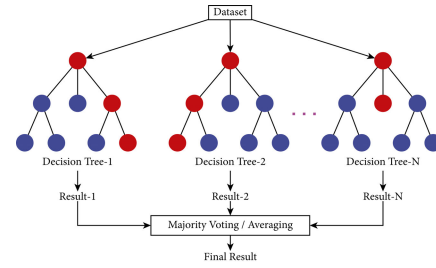


FIGURE 6. Scheme of random forest trees.

require a learning rate. However, in contrast, it includes a regularization parameter c .

g: K-NEAREST NEIGHBOR (K-NN)

It is amongst the classical ML algorithm based on lazy learning techniques [79], [88], [89]. It follows the mechanism of calculating the distance (or similarity) between the unseen document with all other documents in the dataset. Followed by taking the votes of k documents where the distance is minimum (or similarity is maximum) for predicting the class of unseen documents. In our experiment, we used the following cosine similarity S_{\cos} as a comparison measure:

$$S_{\cos}(\mathcal{X}_a, \mathcal{X}_b) = \frac{\mathcal{X}_a \cdot \mathcal{X}_b}{\|\mathcal{X}_a\| \|\mathcal{X}_b\|} = \frac{\sum_{i=1}^n \mathcal{X}_{a_i} \mathcal{X}_{b_i}}{\sqrt{\sum_{i=1}^n \mathcal{X}_{a_i}^2} \sqrt{\sum_{i=1}^n \mathcal{X}_{b_i}^2}} \quad (15)$$

3) EXPERIMENTAL SETUP

We executed the experiment for the sentiment and sarcasm classification in Urdu tweets from two perspectives. The foremost is the classification in the SA manner, where the text of the tweet is given as input to the classifier for training the sentiment and sarcasm classification systems. And secondly, for the utilization of mutual CR, alongside the text of the tweet, the sentiment label is also made part of the input for the training sarcasm classifier; and *vice versa*. The mathematical representation of baseline/SA and CR-based ML functions are already given in the introduction section I. In the similar context, the pseudocode (see algorithm 2) differentiates the conventional approach for the ML classification and the proposed one.

For the comprehensive analysis of the machine-learning task, we have created four variations of the dataset. We maintain that the emojis are the concentric part of tweets (as we have seen their counts and distributions in tables 9 and 10) therefore, we catered for their presence and absence in the experimental variation of the datasets. Details of these datasets are provided in table 12.

Instead of k -fold cross-validation, we performed rigorous experiments through the Monte-Carlo sampling method. In comparison, it gives a better approximation of results (by reflecting the philosophies of both) to cross-validations and static train-test split w.r.t the Pareto principle. The classification in the Monte-Carlo sampling method functions

Algorithm 2 Pseudocode for Outlining the Difference Between Conventional ML and Proposed CR-Based Approaches

Require: A data-frame or delimited text file (D), consisting of three columns: Tweets (T), Sentiment_Label (sen), and Sarcasm_Label (sar).

- 1: $shuffle(D)$
- 2: $T \leftarrow vectorize(pre_process(D[T]))$ ▷ Copy the (pre-processed and vectorized) tweets in a separate variable.
- 3: $sen \leftarrow D[sen]$ ▷ Copy the sentiment labels of the respective tweets in a separate variable.
- 4: $sar \leftarrow D[sar]$ ▷ Copy the sarcasm labels of the respective tweets in a separate variable.
- 5: $t \leftarrow 0.8$ ▷ be a floating value in the range (0, 1]. It is a threshold for splitting dataset in training and testing subsets, where it specifically maintains the size of training data.

Conventional ML approach

- A.1: $\langle X_{train}, X_{test}, y_{train}^l, y_{test}^l \rangle \leftarrow split_into_subsets(T, l, t)$ ▷ This is basically sampling function that split a dataset in the train test split. We emphasize on the stratified sampling. $l \in \{sen, sar\}$ and t is the splitting threshold.
- A.2: $model \leftarrow$ instantiate an ML model.
- A.3: $model.train(X_{train}, y_{train}^l)$
- A.4: $y_{pred}^l \leftarrow model.predict(X_{test})$; ▷ Predict the unseen test set.
- A.5: $evaluate(y_{test}^l, y_{pred}^l)$ ▷ We maintained Monte Carlo sampling and evaluation, see figure 7.

Proposed CR-based approach

- B.1: $tar \leftarrow sen$ ▷ We target sentiment classification.
- B.2: $T' \leftarrow \{x[0] \sim x[1] \mid x \in T \nabla sar\}$ ▷ since sentiment classification is targeted; therefore, sar is appended to the tweets in the respective order so that the training will happen in the presence of sarcasm label. The nabla symbol (∇) is introduced to show the zip function $\therefore T \nabla sar \equiv zip(T, sar)$
- B.3: $\langle X_{train}, X_{test}, y_{train}^{tar}, y_{test}^{tar} \rangle \leftarrow split_into_subsets(T', l, t)$ ▷ See line number A.1 *ibid*.
- B.4: $model \leftarrow$ instantiate an ML model.
- B.5: $model.train(X_{train}, y_{train}^{tar})$
- B.6: $y_{pred}^{tar} \leftarrow model.predict(X_{test})$ ▷ See line number A.4 *ibid*.
- B.7: $evaluate(y_{test}^{tar}, y_{pred}^{tar})$ ▷ See line number A.5 *ibid*.

in multiple rounds (see figure 7 for illustration), *viz.*, for the r rounds—provided that $r \geq 2$ —we do: (1) dataset is shuffled, (2) training-testing splits are taken out in a stratified manner, (3) employing train split for ML, (4) evaluation of resulting predictive model on test split, (5) cumulation/aggregation of results in respective variables, (6) go to step 1 for next round, (7) in last, averaging/division of accumulated results by r [90]. In the course of this paper, we set the value of $r = 10$.

TABLE 12. Dataset variation for the experimentation.

Name	Description
D_{BASE}	This is the baseline/vanilla dataset with the pre-processing as discussed above.
D_{TAGS}	This is the D_{BASE} without hashtags.
D_{TEXT}	This is the D_{BASE} without hashtags, emojis, and emoticons. It indicates the employment of text only.
D_{EMOJI}	This is D_{BASE} with the sole information of emojis; created to explore the effectiveness of distant supervision.

To summarize the numbers, there are 2 vectorization techniques, 2 bag-of-words approaches, 7 classifiers, 4 variations of the dataset, and 10 rounds of Monte-Carlo sampling; thus, we performed 1,120 experiments for approximating results for a single suite of sentiment classification. Hence, the total number of experiments counting the 2 SA classifications and 2 experiments based on CR is equal to 4,480.

4) EVALUATION CRITERIA

The evaluation of each experiment follows the statistical measures outlined in table 13 [83]. As the dataset is balanced for sentiment but somewhat imbalanced for sarcasm, we have used weighted statistics to measure the performance, rather than conventional statistics. Mathematically, this can be expressed as follows [79]:

$$\frac{1}{\sum_{l \in \mathcal{L}} |\tilde{y}_l|} \sum_{l \in \mathcal{L}} |\tilde{y}_l| \Phi(\hat{y}_l, \tilde{y}_l) \quad (16)$$

where, \mathcal{L} is the set of classes/labels, \tilde{y} is the actual/true label, \hat{y} is the predicted label, \tilde{y}_l and y_l are respectively all the true labels and predicted labels that have the label l , $|\tilde{y}_l|$ is the number of true labels that have the label l , and $\Phi(\hat{y}_l, \tilde{y}_l)$ computes the any of the evaluation metrics given in table 13 for the true and predicted labels that have the label l . For example, to compute the weighted precision for label/class l we consider: $\Phi(\hat{y}_l, \tilde{y}_l) = \frac{|\hat{y}_l \cap \tilde{y}_l|}{|\hat{y}_l|} \equiv \frac{T_l}{T_l + F_l}$; where l in $F_l/T_l \Rightarrow l \in \mathcal{L} = \{P, N\}$ and T_l is an element of the confusion matrix (see table 14).

We maintain that the T_P i.e., true positive—is the number of records where the actual and predicted labels are positive ($T_P \equiv |\tilde{y}_l \cap \hat{y}_l|$; $l = P$); and likewise, T_N i.e., true negative—is the number of records where the actual and predicted labels are negative ($T_N \equiv |\tilde{y}_l \cap \hat{y}_l|$; $l = N$). F_P and F_N i.e., false positive and false negative respectively. They show the number of wrong predictions, such that F_P is negative in actual but wrongly predicted as positive ($F_P \equiv |\tilde{y}_{-l} \cap \hat{y}_l|$; $l = P$ and $\neg l = N$), whereas F_N is *vice versa*.

There has been much research on sentiment classification in which the F1-score has been used as the main evaluation metric, as demonstrated in works like [18], [38], and [78]. However, since the dataset used in this paper is slightly imbalanced, the authors are placing more emphasis on achieving a higher balanced accuracy [83], which is

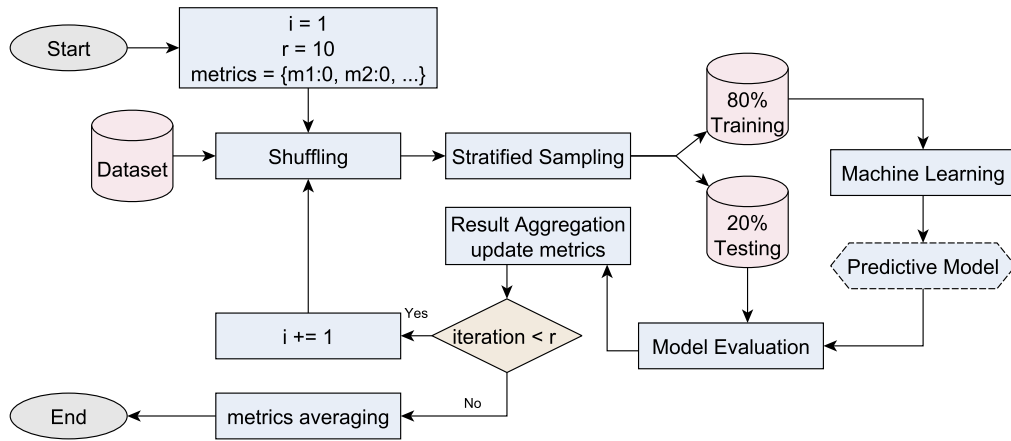


FIGURE 7. Illustration of monte carlo sampling and evaluation.

TABLE 13. Summary of evaluation statistics used in this paper.

Name (abbreviation)	Derivation	Definition/Notes
Precision (P)	$P = \frac{T_P}{T_P + F_P}$	Precision (alternatively known as Positive Predictive Value) reveals the ratio of T_P to the documents that are predicted positively by the Predictive System (PS).
Recall (R)	$R = \frac{T_P}{T_P + F_N}$	Recall (or True Positive Rate) shows the right potential of the PS for predicting positive documents in the subset of all positive documents in the system.
Specificity (S)	$S = \frac{T_N}{T_N + F_N}$	Specificity (or True Negative Rate) is the exact opposite of R . It gives the potential of the PS for negative documents.
F1-score (F)	$F = 2 \cdot \frac{P \cdot R}{P + R}$	F1-score is a harmonic mean of P and R . It is important to use where the dataset is imbalanced; further, it is a strict measure, which has a propensity towards the minima of P and R [90].
Balanced-Accuracy (BA)	$A = \frac{R + S}{2}$	Likewise F , the Balanced-Accuracy is also a mean statistic, which gives an arithmetic mean of R and S .

TABLE 14. A model confusion matrix.

	Actual Labels (\hat{y})	
	N	P
Predicted Labels (\hat{y})	N	F_N
	P	T_P

calculated as the arithmetic mean of the true positive rate and true negative rate [91], [92], [93].

IV. RESULTS AND DISCUSSION

The results and discussion on it are organized into five subsections, each focusing on specific aspects of the study. Subsections IV-A to IV-D provide individual analyses of the SA and CR-based sentiment and sarcasm classifications, respectively. These sections discuss the performance of each approach, highlighting potential errors through the presentation of examples that represent common mistakes across different datasets and machine learning models. A comprehensive review of these errors is also provided.

The final subsection, namely IV-E, presents a comparative analysis between the SA and CR-based approaches for sentiment and sarcasm classification. This section evaluates the extent of improvement achieved by the CR-based

approach compared to its SA counterpart. Throughout these subsections, the primary focus of discussion is placed on the BA statistic, which serves as a key measure of performance. For a more detailed account of the datasets, data processing techniques, and evaluation metrics used in the study, readers are referred to the appendices.

A. STAND-ALONE SENTIMENT CLASSIFICATION

The overall result of SA classification is, in some measure, fair—as the averaged BA runs between ≈ 59.35 – 67.38% in all of the experiments catering for all features selection, vectorization, and datasets. *Viz.*, it shows, that there is ≈ 6.68 – 14.71% improvement w.r.t the baseline distribution of sentiment in the dataset (since 52.67% records out of total and w.r.t the sentiment classification, are negative (*cf.* table 2); therefore, if any of the classifiers marks all test documents negative then we can achieve at least 52.67% accuracy. Hence, the reported gain is calculated for such a baseline).

We observed that TF-IDF vectorization is conducive to achieving better results than count vectorization. However, uni-gram and n -gram showed almost similar results except for the dataset D_{TEXT} (i.e., the dataset where we eliminated all emojis, emoticons, hashtags), where uni-gram has got a trivial improvement. In the comparative analysis of ML classifiers, we found that classifiers that are probabilistic and

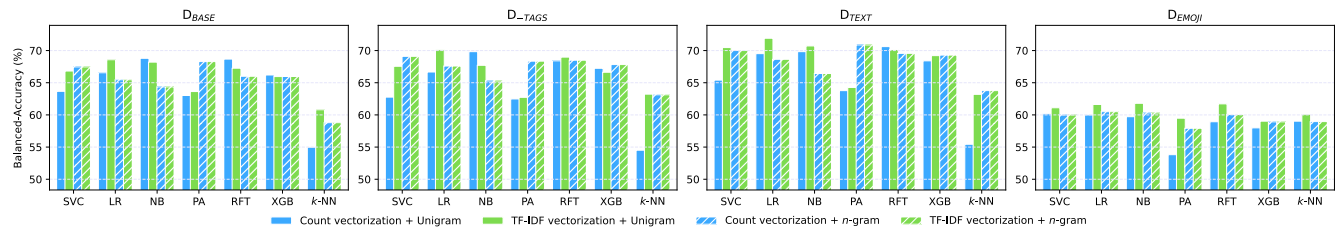


FIGURE 8. Balanced-accuracy achieved on datasets for the SA sentiment classification.

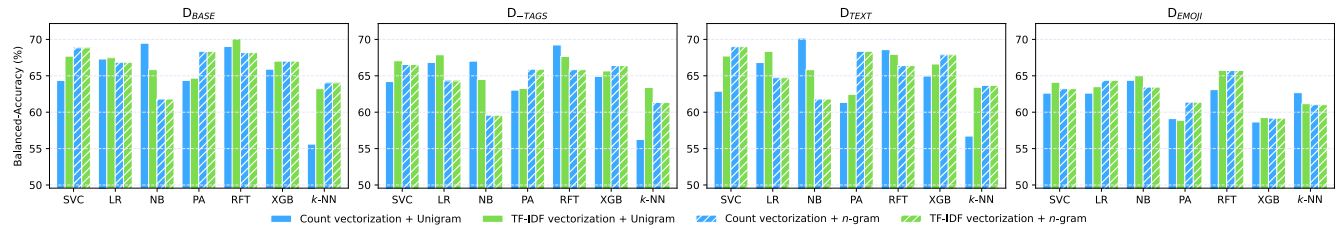


FIGURE 9. Balanced-accuracy achieved on datasets for the SA sarcasm classification.

statistical such as NB, LR, and RF are the most advantageous for SA sentiment classification. In the comparative analysis of datasets with a collective impact created therein (where the impact is considered with the averaged BA of experiments employed therein the dataset), we found the D_{TEXT} secured the highest result (i.e., $\approx 67.38\%$ averaged BA); whereas, the dataset D_{EMOJI} (i.e., the dataset with only emojis; forbye it, the very dataset is prepared to investigate the effectiveness of distant supervision technique) appeared to be very unsuccessful in discriminating positive and negative tweets by attaining $\approx 59.35\%$ averaged BA. Hence, as the SA approach for the USA, we maintain that the usage of emojis is not a very conducive approach for data labelling and data creation.

In a similar context, the BA achieved in the experiments w.r.t classifiers at the individual dataset level are maintained in figure 8; where the highest accuracy is: 69.45% in D_{BASE} with RF+count vectorization+n-gram, 70.14% in D_{TAGS} with LR+TF-IDF+uni-gram, 71.91% in D_{TEXT} with LR+TF-IDF+uni-gram, and 61.81% in D_{EMOJI} with NB+TF-IDF+uni-gram.

We performed a detailed analysis of the misclassifications in the SA sentiment analysis. The conclusion of these mistakes is itemized below; however, for a better understanding, the list of examples is also maintained.

Example 1. Urdu Tweet: باچا خان کی روہ [sic.] سے بے حد معذرت
Translation: *Extremely sorry with the soul of Bacha Khan*
. Original Labels: *Negative+Non-Sarcastic*.

Example 2. Urdu Tweet: تم کہو اور میں جان دے دوں؟؟ معذرت میں اب
 وہ نہیں رہا!!! Translation: *You say and I will give (sacrifice) my life ?? Sorry, I am not that anymore !!!* Original Labels: *Negative+Sarcastic*.

Example 3. Urdu Tweet: آج پہلی بار اتنے غصے میں دیکھا آپ کو ورنہ سبھی سے
پیار سے بات کرتے ہیں آپ خیر میں اپنے الفاظ پر معذرت خواہ ہوں #سوری

Translation: *Today, for the first time, I saw you in such a rage, otherwise, you would talk to everyone with love* Well, I apologize for my words #sorry. Original Labels *Positive+Non-Sarcastic*.

Example 4. Urdu Tweet: شوہر نے بیوی کا حال چال جاننے کے لیے وٹس ایپ پر لکھا۔ کیسا ہے سر درد* پر غلطی سے ٹائپ ہو گیا کیسی ہو سر درد* شوہر کا آفس ختم ہوئے ایک گھنٹہ ہو گیا پر گھر نہیں جا رہا شوہر کا نام ہو چھتے کی ضرورت نہیں #kahani Translation: *Husband wrote on WhatsApp to know his wife's condition. How is (your) headache? ** Mistakenly a typo occurred as 'How are you headache?' * #kahani. Original Labels: *Positive+Sarcastic*

Following is the review of errors in SA sentiment classification.

- The utilization of the term *معذرت* /məʔzərət/ (transl. 'sorry', also encompassing the connotations of excuse, 'apology', or 'plea') exhibits a multifaceted nature, as evidenced by the first three examples. In the first example, it conveys feelings of sorrow and grief, while in the second example, it sarcastically expresses an excuse. The third example employs it to seek genuine remorse. Consequently, it is apparent that tweets devoid of contextual clues regarding sarcasm may lead to erroneous categorization. Similarly, the term *جان* /dʒɑːn/ (transl. 'soul', 'energy', 'beloved', etc.) necessitates a contextual framework to ensure precise classification.
- Humorous expressions, as exemplified in the final instance, often employ wordplay, such as *گھنٹہ* /'gʰɛntə(:)/ (transl. 'clock', also used as a slang term for 'nothing'). Furthermore, the tweet itself incorporates multi-word expressions like *سر درد* /sər dər d/ (transl. 'headache'), which may introduce an element of ambiguity and potential inaccuracies.

- Lastly, the integration of emojis in conjunction with textual content within the respective tweets does not consistently align with precise categorization. Consequently, the reliance solely on emojis is inadequate for accurate classification in the absence of contextual information.

B. STAND-ALONE SARCASM CLASSIFICATION

The SA sarcasm classification appears better than the SA sentiment classification. The averaged BA, in all experiments conducted for sarcasm classification, is found running between ≈ 62.34 – 65.80% . *Viz.*, which means, there is 6.57–10% improvement w.r.t the baseline distribution of sarcastic tweets (i.e., 55.77%) of the whole corpus.

We observed that normally count-based vectorization performed better than TF-IDF vectorization, except for the dataset $D_{\text{-TAGS}}$ (i.e., the dataset where hashtags were eliminated) with n -gram features. Moreover, on average, uni-gram and n -gram appeared similar in all datasets. In the comparative analysis, we found NB and RF are the most successful ML classifiers for the SA sarcasm classification; this is akin to the sentiment classification.

In the comparative analysis of datasets, we found that maximum impact, w.r.t the averaged BA, is created in D_{BASE} i.e., 65.8% whereas D_{EMOJI} appears to have the least impact with 62.34% average BA. Since the sole existence of emojis, without the information of words or context, is not producing any better results, therefore, as for the SA approach, we maintain that it is also not a very conducive approach for sarcasm classification or dataset labelling in a distant supervision fashion. Moving onwards, the BA achieved in the experiments w.r.t classifiers at dataset level is maintained in figure 9; where the highest accuracy is: 70.12% in D_{BASE} with RF+TF-IDF+uni-gram, 69.24% in $D_{\text{-TAGS}}$ with RF+Count+uni-gram, 70.18% in D_{TEXT} with NB+Count+uni-gram, and 65.75% in D_{EMOJI} with RF+TF-IDF+uni-gram. The error analysis on the SA sarcasm classification concluded in the following points with examples.

Example 5. Urdu Tweet: جو لڑکیاں مجھے انکیس [sic] میں میج کر رہی ہیں۔ ان سب بھائیوں سے معذرت میں بھی مرد ہوں۔ #MeToo Translation: *The girls who are texting me in the inbox. Sorry to all these brothers. I am also a man.* #MeToo Original Labels: Positive+Sarcastic

Example 6. Urdu Tweet: اس مرد میدان کو میں سلام پیش کرتا ہوں اپوزیشن کے بیٹے بجا دی تو نے سر سلوٹ۔ Translation: *Salute to that valorous man, you loused up the opposition, salute to you sir.* Original Labels: Positive+Non-Sarcastic

Example 7. Urdu Tweet: معذرت لیکن نانا کہلانے کے لیے پہلے اولاد کو تسلیم کرنا ضروری ہوگا۔ Translation: *Sorry, but to be called Nana (maternal grandfather), it is necessary to admit the first child.* Original Labels: Negative+Sarcastic

Example 8. Urdu Tweet: پٹواری ایک دوسرے کو ری ٹویٹ کرتے ہوئے!۔ lol Translation: *Patwaris, re-tweeting each other! lol.* Original Labels: Negative+Non-Sarcastic

Following is the review of errors in SA sarcasm classification.

- Example 5 demonstrates the nuanced usage of the hashtag #MeToo, which typically denotes discussions surrounding sensitive topics such as physical/sexual abuse, bullying, and persecution. However, it is often employed as a tool for sarcasm. Consequently, its interpretation without proper contextual cues may result in misclassification.
- In the case of example 6, the word سلام /sə.lɑ:m/ (literally meaning ‘peace’, commonly used as a salutation or greeting) shares a comparable characteristic with the word معذرت. Its presence in a positive tweet signifies a non-sarcastic context, whereas its absence indicates the opposite. Furthermore, example 7 underscores the necessity of a comprehensive textual analysis to accurately identify sarcastic elements.
- Additionally, certain words are specifically designed for mockery, such as پٹواری /pə.t.vɑ:.ri:/ (literally referring to a ‘village accountant’ and in the realm of Pakistani politics, denoting a ‘member/supporter of the Pakistan Muslim League (Nawaz Sharif Group)’). Such words often lead to misclassification as sarcastic. However, in example 8, the absence of a sarcastic pivot within the text indicates a non-sarcastic usage of پٹواری.

C. CR-BASED SENTIMENT CLASSIFICATION

The addition of sarcastic and sentiment-based cognition in the process turns out to be very productive for the improvement of classification accuracy. The averaged BA of all experiments conducted for CR-based sentiment classification runs between ≈ 73 – 74.5% . It means, there is ≈ 20.33 – 22% improvement over baseline sentiment distribution.

Throughout the datasets, we found that the performance of count and TF-IDF vectorization is approximately similar; however, n -gram shows a cursory improvement in comparison with uni-gram features. For the ML classifiers, we found XGB as an outperformer followed by RF and LR. Lastly, figure 10 shows the BA achieved by ML classifiers in the datasets; where the highest accuracy is: 77.97% in D_{BASE} with XGB+TF-IDF for both uni-gram and n -gram features, 78.13% in $D_{\text{-TAGS}}$ with XGB+TF-IDF+uni-gram, 77.97% in D_{TEXT} with RF+Count+uni-gram, and 79.17% in D_{EMOJI} with XGB+TF-IDF for both uni-gram and n -gram features.

In general, the misclassified tweets with CR-based sentiment classification are not inclusive of what happened in the SA classification. The error analysis and conclusion thereof are provided subsequently with the example.

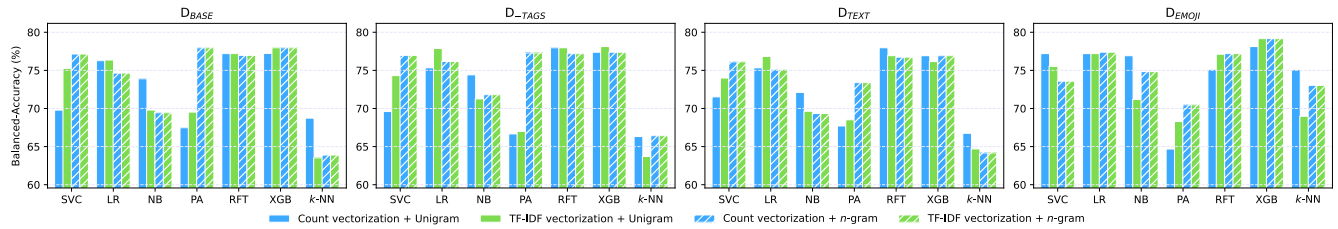


FIGURE 10. Balanced-accuracy achieved on datasets for sentiment classification utilizing CR based on sarcasm information.

Example 9. Urdu Tweet: *Marriage* ویسے آپس کی بات ہے کہ *Shaadi Kar Ke Jeo* 🤪🤪🤪 #ShaadiKarKeJeo 🤪🤪🤪 *Lot* می پاپا کہنے والے آھی جاتے ہیں 🤪🤪🤪 #LOLSurprise Translation: *Look bro, it is between us that the marriage—be it is arranged or love, Mummy Papa callers would do arrive (born)* 🤪🤪🤪 #ShaadiKarKeJeo 🤪🤪🤪 #LOLSurprise 🤪🤪🤪 Original Labels: *Positive+Non-Sarcastic*

Example 10. Urdu Tweet: *اور کتنا نام پر اٹھے* 😞 اور کتنا ناام پر اٹھے 😞 ایک گھنٹہ تو پیچھے رہ گیا ہے 😞 اور کتنا ناام پر اٹھے 😞 Translation: *One hour is left behind 😞 how exactly on time one should up? 😞* Original Labels: *Positive+Sarcastic*

Example 11. Urdu Tweet: *عشق شاعری #سوج کا سفر معذرت خواہ ہیں*... Translation: *#love_poetry #journey_of_thought I'm sorry... O' heart handed you over to the worthless* Original Labels: *Negative+Non-Sarcastic*

Example 12. Urdu Tweet: *اسے کہتے ہیں کھیانی لی کہا نوچے* 😞 جب جواب نہ ہو معذرت خوانہ [sic] اور مدافعانہ رویہ اپنانے کی گئی نکلنا بہتر ہوتا ہے اور بیک میلمانیا Translation: *That is called showing anger after being embarrassed 😞 When there is no answer, it is better to apologize and take a defensive stance, and this is what the blackmailer mafia did 😞* Original Labels: *Negative+Sarcastic*

Following is the review of errors in CR-based sentiment analysis.

- A significant number of errors arise when the content is composed in a code-mixed or code-switched style. Example 9 serves as an illustration of such behavior.
- The presence of short tweets or fragmented/incomplete sentences contributes to misclassification. This can be observed in example 10, which exhibits the aforementioned pattern.
- Additionally, the inclusion of poetic expressions (see example 11) within tweets introduces further confusion. This is particularly true when a philosophical content such as hashtags resembling *عشق شاعری* #عشق_شاعری (/ʃɪq fʌ:ʃ.rɪ:/ (transl. ‘love poetry’) and *سوج کا سفر* #سوج_کا_سفر (/so:tʃ ka sə.fər/ (transl. ‘journey of thought’) are combined with heart-breaking or loving emojis such as 🤪.
- Tweets that predominantly consist of positive phrases are more susceptible to misclassification. For instance, example 12 encompasses phrases like *معذرت خوانہ* [sic.] *معذرت خواہانہ* /məʔzərətʃ xɑ:hɑ:na / (transl. ‘apologetic’),

مدافعانہ رویہ اپنانے /mudʌ:fɛʔɑ:nɑ: rɔvɪjɑ: əpnɑ: ke/ (transl. ‘embracing a defensive attitude’), and بہتر ہوتا ہے /bɛfɪ.təʃ hɔ:tɑ: hæ:/ (literal transl. ‘is better’). Such phrases may overshadow the overall narrative of the tweet.

D. CR-BASED SARCASM CLASSIFICATION

Likewise, in CR-based sentiment classification, we observed a significant improvement in CR-based sarcasm classification, with the yield of ≈ 72.57 – 76.30 % averaged BA on all of the experiments conducted throughout the course of respective work. Compared to the baseline distribution of sarcastic tweets, the said averaged BA shows the improvement between ≈ 20 – 23.6 %.

We observed that, in most of the experiments, the performance of TF-IDF-based vectorization and *n*-gram appear to be better techniques in comparison to their respective counterparts. In the comparative analysis of ML classifiers, we found that RF and XGB are more advantageous with the uni-gram and *n*-gram features respectively. Lastly, figure 11 shows the BA achieved by ML classifiers in the datasets; where the highest accuracy is: 78.85% in D_{BASE} with RF+Count+uni-gram, 77.89% in D_{TAGS} with XGB+TF-IDF for both uni-gram and *n*-gram features, ≈ 79 % in D_{TEXT} with RF+Count+uni-gram, and 80.74% in D_{EMOJI} with XGB+TF-IDF+uni-gram.

The subsequent examples provide the survey of errors in CR-based sarcasm analysis. The conclusion thereof is itemized below.

Example 13. Urdu Tweet: *میں ہنس کے کاٹ لیتی ہوں کچن میں سبزیاں*... Translation: *I happily cut all the vegetables in the kitchen but when I cut onions, my eyes get wet* 🤪🤪🤪 Original Labels: *Negative+Non-Sarcastic*

Example 14. Urdu Tweet: *نہ جانے کون سے #ویٹامن ہے موبائل میں*... Translation: *Don't know which #vitamin is in mobile* 🤪🤪🤪 *If I don't use it for an hour, I start feeling weak* 🤪🤪🤪 Original Labels: *Positive+Sarcastic*

Example 15. Urdu Tweet: *آنٹی جی نثر شاعری ہے، اور پھر آزاد شاعری کا*... Translation: *Dear auntie, this is a prose poetry, and then seasoning of free-verse poetry* 🤪🤪🤪 Original Labels: *Positive+Non-Sarcastic*

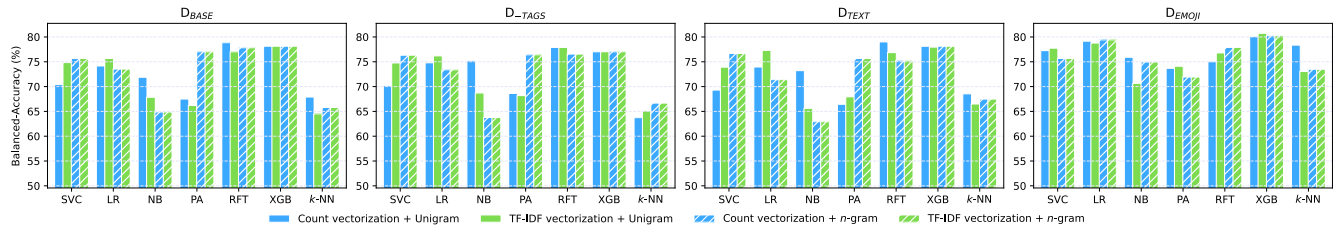


FIGURE 11. Balanced-Accuracy achieved on datasets for sarcasm classification utilizing CR based on sentiment information.

Example 16. Urdu Tweet: سلام خلوص محبت عقیدت احترام دعائیں و پیار ہے کئی ایسی کون مل گئی ہے #اچھی ایسی اچھی لگتی ہے بہت #راس نہیں آتی ہے بے چاہتی تو ہے بہت پر #پاس نہیں آتی ہے معذرت کے ساتھ شام بخیر زندگی
 Translation: Greetings, sincere love, devotion, respect, prayers and love (person who's) Humble How can be a tyrant? Heart! why do you've such restlessness you found whom #[untranslatable word] (though) seems pretty but doesn't appear #suitable (though she) likes a lot but doesn't #come_closer with due respect good evening
 Original Label: Negative+Sarcastic

Following is the review of errors in CR-based sarcasm classification.

- The observed errors in sarcasm classification within sentiment analysis (SA) have been minimized to a negligible extent. However, a significant number of errors encountered in context-based classification arise from the complex interplay of emojis, the tweets' brevity, and puns' presence.
- Tweets that engender confusion in terms of their content, such as those blurring the lines between real and fabricated experiences (see example 13), evoking thoughtfulness (see example 14), or loose talking and double meanings (see example 15), possess a higher likelihood of being misclassified.
- Both short and excessively long tweets, which attempt to encompass a wide range of subjects and inundate the reader with an abundance of emojis (see example 16), are prone to misclassification.

Table 15 provides the sample of correct prediction with XGB+TF-IDF+n-gram; tweets alongside the class probabilities are also maintained for SA and CR-based classification.

	D _{BASE}								D _{TAGS}								D _{TEXT}								D _{EMOJI}							
Count + unigram	6.1	9.7	5.1	4.5	8.5	11	14	-	6.8	8.7	4.6	4.2	9.6	10	12	-	6.1	5.8	2.3	3.9	7.4	8.5	11	-	17	17	17	11	16	20	16	
Count + n-gram	9.7	8.8	6.9	11	7.8	11	13	-	11	9.7	5.2	11	9.6	10	8.6	-	8.7	6.9	4.2	4	5.7	7.8	18	-	16	18	16	4.3	18	20	17	
TF-IDF + unigram	8.4	7.8	1.6	5.9	9.9	12	2.7	-	6.8	7.7	3.6	4.3	9	11	0.48	-	3.5	4.9	-1.1	4.2	6.8	6.9	1.5	-	14	16	9.4	8.8	15	20	8.9	
TF-IDF + n-gram	9.7	9.2	5.1	9.7	11	12	5.1	-	7.9	8.6	6.5	8.9	8.7	9.6	3.3	-	6.1	6.5	2.9	2.5	7.2	7.6	0.39	-	14	17	15	13	17	20	14	
	SVC	LR	NB	PA	RFT	XGB	k-NN	SVC	LR	NB	PA	RFT	XGB	k-NN	SVC	LR	NB	PA	RFT	XGB	k-NN	SVC	LR	NB	PA	RFT	XGB	k-NN				

FIGURE 12. Improvement in sentiment classification achieved through using CR based on sarcasm.

The CR-based result clearly shows improvement in comparison to the SA.

E. COMPARATIVE ANALYSIS

Figures 12 and 13 present a comparative analysis of the improvements achieved through CR-based sentiment and sarcasm classification over their SA counterparts, showcasing the results for each feature selection pair across different classifiers. The color gradient ranging from gray to bright green in the cells indicates the degree of improvement, with brighter shades denoting higher improvement.

Regarding sentiment classification (refer to Figure 12), a consistent positive trend of improvement was observed in all experiments, except for a single case where a negative improvement of -1.1% was recorded in D_{TEXT} with the NB+TF-IDF+uni-gram experiment. Notably, the most remarkable improvement was observed in D_{EMOJI}, exhibiting an impressive approximate improvement of 15.2±3.7σ% compared to SA sentiment classification. Furthermore, the dataset-wise percentage improvements of CR-based sentiment classification over its SA counterpart are as follows: 8.47±3.2σ% in D_{BASE}, 7.76±3σ% in D_{TAGS}, and 5.72±3.8σ% in D_{TEXT}. Therefore, based on these statistics, it can be inferred that sentiment classification relying solely on D_{TEXT} (i.e., excluding hashtags, emojis, and emoticons) is an unreliable technique, as it exhibits lower improvement rates and higher standard deviations. Among the ML classifiers, the ensemble techniques XGB and RF demonstrated the maximum improvements, followed by LR in the third position. Conversely, NB and PA proved to be less effective classifiers for CR-based sentiment classification. The overall collective improvement in CR-based sentiment classification, computed as an average of improvements, amounts to approximately 9.3%.

hypothesis in our experiments; the dataset where the emojis are removed performed better than the dataset that solely relies on the emojis; however, when the CR is exerted, the emoji-based dataset improved to be the equivalent of its counterparts.

After comprehensive experiments on a dataset (extracted from Twitter), feature selection, and ML classifiers, we conclude that the CR-based proposed approach outperforms the SA classification by showing $\approx 9.3\%$ and $\approx 9.1\%$ improvement respectively for sentiment and sarcasm classification; alongside yielding $\approx 21 \pm 1\%$ improvement over the baseline distributions. Hence, the CR-based proposed technique is more conducive for sarcasm and sentiment classification.

In our future work, we have identified several areas of interest that we plan to explore further. Firstly, we aim to expand our dataset by incorporating more code-mixed and code-switched data. This expansion will allow us to better capture the linguistic phenomena and challenges associated with language mixing. Secondly, we plan to explore feature enhancement techniques based on distributional semantics. These techniques leverage the semantic relationships and contextual information embedded in large corpora to enhance

the representation of text. By incorporating these techniques into our models, we anticipate improved performance in sentiment and sarcasm classification tasks. Lastly, we are eager to experiment with deep neural network (DNN) architectures. DNNs have demonstrated remarkable success in various natural language processing tasks, and we believe they have the potential to enhance the performance of sentiment and sarcasm classification models as well. By leveraging the expressive power of DNNs, we aim to capture more intricate patterns and nuances in textual data, ultimately leading to improved classification accuracy.

**APPENDIX A
DETAILED RESULTS**

Individual results for the SA sentiment classification on D_{BASE} , D_{TAGS} , D_{TEXT} , and D_{EMOJI} are respectively provided in tables 16–19. Similarly, tables 20–23 provide individual results for SA sarcasm classification. Tables 24–27 provide individual results for CR-based sentiment classification. And tables 28–31 provide individual results for CR-based sarcasm classification. All of the numbers reported in the appendix are the percentages. We have marked the highest values in bold letters.

TABLE 16. Results of SA sentiment classification on dataset D_{BASE} .

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.5498	.5250	.5596	.5392	.5603	.5146	.5118	.5162	.5196	.5097
	TF-IDF	.6080	.6080	.6086	.6078	.6082	.5880	.5886	.5893	.5882	.5877
Linear SVC	Count	.6366	.6374	.6377	.6373	.6359	.6730	.6734	.6737	.6733	.6726
	TF-IDF	.6682	.6695	.6695	.6699	.6665	.6746	.6757	.6825	.6796	.6696
Logistic Regression Class.	Count	.6655	.6667	.6667	.6667	.6644	.6918	.6929	.6928	.6931	.6906
	TF-IDF	.6858	.6874	.6902	.6893	.6822	.6547	.6518	.6762	.6634	.6460
Naïve Bayes	Count	.6879	.6892	.6967	.6931	.6827	.6586	.6570	.6604	.6569	.6603
	TF-IDF	.6820	.6813	.7003	.6893	.6746	.6435	.6368	.6803	.6569	.6302
Passive Aggressive	Count	.6302	.6311	.6311	.6311	.6293	.6662	.6667	.6667	.6667	.6656
	TF-IDF	.6363	.6371	.6370	.6373	.6354	.6829	.6851	.6860	.6863	.6795
Random Forests	Count	.6867	.6883	.6893	.6893	.6841	.6945	.6955	.6961	.6961	.6928
	TF-IDF	.6725	.6728	.6799	.6765	.6685	.6600	.6560	.6807	.6667	.6533
XG Boost	Count	.6620	.6643	.6667	.6667	.6574	.6663	.6678	.6704	.6699	.6627
	TF-IDF	.6596	.6612	.6637	.6634	.6558	.6597	.6614	.6694	.6667	.6528

TABLE 17. Results of SA sentiment classification on dataset D_{TAGS} .

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.5451	.5352	.5491	.5392	.5511	.5296	.5096	.5390	.5392	.5201
	TF-IDF	.6323	.6334	.6334	.6337	.6309	.6313	.6328	.6331	.6337	.6289
Linear SVC	Count	.6276	.6276	.6282	.6275	.6278	.6574	.6572	.6586	.6569	.6580
	TF-IDF	.6756	.6763	.6763	.6765	.6747	.6910	.6934	.6974	.6961	.6859
Logistic Regression Class.	Count	.6667	.6669	.6677	.6667	.6667	.6824	.6832	.6832	.6832	.6816
	TF-IDF	.7014	.7038	.7067	.7059	.6969	.6759	.6751	.7008	.6863	.6656
Naïve Bayes	Count	.6983	.6999	.7059	.7030	.6936	.6645	.6635	.6657	.6634	.6656
	TF-IDF	.6771	.6775	.6964	.6863	.6679	.6528	.6453	.6964	.6667	.6389
Passive Aggressive	Count	.6248	.6239	.6260	.6238	.6258	.6599	.6603	.6606	.6602	.6595
	TF-IDF	.6273	.6277	.6285	.6275	.6272	.6835	.6844	.6874	.6863	.6807
Random Forests	Count	.6840	.6858	.6858	.6863	.6818	.6840	.6858	.6858	.6863	.6818
	TF-IDF	.6898	.6920	.6992	.6961	.6836	.6849	.6837	.7073	.6931	.6768
XG Boost	Count	.6725	.6747	.6763	.6765	.6684	.6733	.6740	.6781	.6765	.6701
	TF-IDF	.6663	.6678	.6704	.6699	.6627	.6782	.6796	.6928	.6863	.6702

TABLE 18. Results of SA sentiment classification on dataset D_{TEXT}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.5544	.5459	.5583	.5490	.5598	.5033	.4870	.5054	.5149	.4918
	TF-IDF	.6319	.6340	.6369	.6373	.6266	.6378	.6382	.6458	.6436	.6320
Linear SVC	Count	.6541	.6537	.6550	.6535	.6547	.6852	.6863	.6863	.6863	.6841
	TF-IDF	.7047	.7055	.7057	.7059	.7034	.7002	.7027	.7083	.7059	.6946
Logistic Regression Class.	Count	.6952	.6959	.6959	.6961	.6944	.7037	.7054	.7055	.7059	.7015
	TF-IDF	.7191	.7210	.7243	.7228	.7155	.6863	.6865	.7094	.6961	.6766
Naïve Bayes	Count	.6983	.6999	.7059	.7030	.6936	.6739	.6735	.6748	.6733	.6746
	TF-IDF	.7072	.7087	.7265	.7157	.6987	.6644	.6608	.6979	.6765	.6522
Passive Aggressive	Count	.6377	.6376	.6390	.6373	.6382	.6824	.6832	.6832	.6832	.6816
	TF-IDF	.6427	.6436	.6436	.6436	.6418	.7087	.7105	.7151	.7129	.7046
Random Forests	Count	.7062	.7077	.7089	.7087	.7036	.7130	.7149	.7154	.7157	.7102
	TF-IDF	.7016	.7018	.7119	.7059	.6973	.6956	.6954	.7231	.7059	.6853
XG Boost	Count	.6840	.6858	.6858	.6863	.6818	.6908	.6923	.6928	.6931	.6886
	TF-IDF	.6921	.6945	.6962	.6961	.6882	.6929	.6938	.6982	.6961	.6898

TABLE 19. Results of SA sentiment classification on dataset D_{EMOJI}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.5903	.5906	.5975	.5980	.5825	.5799	.5794	.5873	.5882	.5715
	TF-IDF	.6011	.5953	.6145	.6078	.5943	.5896	.5907	.5930	.5941	.5852
Linear SVC	Count	.6019	.6035	.6068	.6078	.5959	.5856	.5876	.5874	.5882	.5831
	TF-IDF	.6111	.6126	.6173	.6176	.6046	.6000	.6014	.6031	.6040	.5961
Logistic Regression Class.	Count	.6000	.6014	.6031	.6040	.5961	.6000	.6014	.6031	.6040	.5961
	TF-IDF	.6162	.6168	.6219	.6214	.6111	.6051	.6040	.6135	.6117	.5985
Naïve Bayes	Count	.5972	.5939	.6118	.6078	.5866	.6065	.6025	.6242	.6176	.5953
	TF-IDF	.6181	.6170	.6315	.6275	.6087	.6032	.5988	.6173	.6117	.5947
Passive Aggressive	Count	.5382	.5394	.5398	.5392	.5372	.5737	.5744	.5747	.5743	.5731
	TF-IDF	.5949	.5970	.5970	.5980	.5918	.5792	.5805	.5813	.5825	.5758
Random Forests	Count	.5894	.5908	.5912	.5922	.5865	.5801	.5816	.5816	.5825	.5777
	TF-IDF	.6172	.6183	.6211	.6214	.6130	.6007	.6017	.6074	.6078	.5935
XG Boost	Count	.5799	.5536	.6281	.5980	.5617	.5788	.5498	.6259	.5941	.5636
	TF-IDF	.5903	.5675	.6393	.6078	.5727	.5892	.5638	.6372	.6040	.5745

TABLE 20. Results of SA sarcasm classification on dataset D_{BASE}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.5561	.4742	.6010	.5196	.5927	.5157	.3316	.6179	.4653	.5660
	TF-IDF	.6323	.6233	.6408	.6238	.6409	.6404	.6385	.6457	.6373	.6434
Linear SVC	Count	.6437	.6445	.6475	.6436	.6437	.6770	.6799	.6803	.6796	.6743
	TF-IDF	.6770	.6799	.6803	.6796	.6743	.6885	.6966	.7047	.7030	.6740
Logistic Regression Class.	Count	.6731	.6768	.6773	.6765	.6697	.6819	.6863	.6863	.6863	.6775
	TF-IDF	.6749	.6829	.6844	.6863	.6634	.6687	.6743	.7124	.6931	.6442
Naïve Bayes	Count	.6947	.7027	.7046	.7059	.6836	.6657	.6608	.6715	.6602	.6712
	TF-IDF	.6585	.6649	.7054	.6863	.6307	.6181	.6101	.7093	.6569	.5794
Passive Aggressive	Count	.6437	.6445	.6475	.6436	.6437	.6731	.6768	.6773	.6765	.6697
	TF-IDF	.6468	.6482	.6512	.6471	.6465	.6836	.6921	.6948	.6961	.6712
Random Forests	Count	.6903	.6967	.6977	.6990	.6816	.6725	.6813	.6850	.6863	.6588
	TF-IDF	.7012	.7103	.7166	.7157	.6867	.6819	.6894	.7153	.7030	.6609
XG Boost	Count	.6591	.6682	.6765	.6765	.6417	.6641	.6716	.6885	.6832	.6450
	TF-IDF	.6702	.6793	.6862	.6863	.6541	.6702	.6793	.6862	.6863	.6541

TABLE 21. Results of SA sarcasm classification on dataset D_TAGS.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.5626	.4939	.6010	.5294	.5957	.5216	.3583	.5989	.4706	.5727
	TF-IDF	.6339	.6282	.6410	.6275	.6404	.6135	.6181	.6186	.6176	.6093
Linear SVC	Count	.6421	.6471	.6471	.6471	.6372	.6684	.6750	.6748	.6765	.6604
	TF-IDF	.6707	.6771	.6779	.6796	.6617	.6655	.6745	.6909	.6863	.6447
Logistic Regression Class.	Count	.6684	.6750	.6748	.6765	.6604	.6749	.6829	.6844	.6863	.6634
	TF-IDF	.6789	.6883	.6977	.6961	.6618	.6430	.6447	.6922	.6699	.6161
Naïve Bayes	Count	.6702	.6793	.6862	.6863	.6541	.6667	.6677	.6708	.6667	.6667
	TF-IDF	.6450	.6481	.7041	.6765	.6136	.5959	.5807	.6896	.6373	.5546
Passive Aggressive	Count	.6304	.6340	.6345	.6337	.6271	.6661	.6719	.6717	.6733	.6589
	TF-IDF	.6328	.6369	.6366	.6373	.6283	.6591	.6682	.6765	.6765	.6417
Random Forests	Count	.6924	.7012	.7055	.7059	.6789	.6841	.6921	.7108	.7030	.6653
	TF-IDF	.6766	.6860	.7003	.6961	.6571	.6585	.6649	.7054	.6863	.6307
XG Boost	Count	.6493	.6555	.6765	.6699	.6287	.6433	.6511	.6717	.6667	.6199
	TF-IDF	.6567	.6657	.6785	.6765	.6370	.6641	.6716	.6885	.6832	.6450

TABLE 22. Results of SA sarcasm classification on dataset D_TEXT.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.5673	.4940	.6132	.5347	.5999	.5246	.3508	.6478	.4752	.5740
	TF-IDF	.6343	.6033	.6625	.6139	.6548	.6368	.6106	.6598	.6176	.6560
Linear SVC	Count	.6289	.6317	.6328	.6311	.6268	.6486	.6511	.6522	.6505	.6466
	TF-IDF	.6772	.6823	.6821	.6832	.6712	.6901	.6994	.7071	.7059	.6742
Logistic Regression Class.	Count	.6683	.6729	.6726	.6733	.6632	.6749	.6792	.6790	.6796	.6701
	TF-IDF	.6836	.6921	.6948	.6961	.6712	.6474	.6524	.6963	.6765	.6183
Naïve Bayes	Count	.7018	.7092	.7126	.7129	.6907	.6801	.6776	.6850	.6765	.6838
	TF-IDF	.6585	.6649	.7054	.6863	.6307	.6181	.6101	.7093	.6569	.5794
Passive Aggressive	Count	.6135	.6181	.6186	.6176	.6093	.6556	.6578	.6598	.6569	.6542
	TF-IDF	.6246	.6282	.6294	.6275	.6217	.6836	.6921	.6948	.6961	.6712
Random Forests	Count	.6860	.6935	.6944	.6961	.6759	.6817	.6892	.6921	.6931	.6704
	TF-IDF	.6796	.6874	.6933	.6931	.6661	.6641	.6716	.6885	.6832	.6450
XG Boost	Count	.6498	.6553	.6708	.6667	.6330	.6552	.6627	.6759	.6733	.6371
	TF-IDF	.6663	.6741	.6856	.6832	.6494	.6789	.6883	.6977	.6961	.6618

TABLE 23. Results of SA sarcasm classification on dataset D_EMOJI.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.6269	.6286	.6317	.6275	.6264	.6140	.6086	.6211	.6078	.6202
	TF-IDF	.6117	.6091	.6177	.6078	.6155	.6103	.6130	.6161	.6117	.6090
Linear SVC	Count	.6262	.6359	.6370	.6408	.6117	.6421	.6471	.6471	.6471	.6372
	TF-IDF	.6411	.6503	.6504	.6535	.6288	.6324	.6411	.6407	.6436	.6212
Logistic Regression Class.	Count	.6262	.6359	.6370	.6408	.6117	.6423	.6495	.6490	.6505	.6342
	TF-IDF	.6351	.6433	.6440	.6471	.6231	.6439	.6523	.6541	.6569	.6309
Naïve Bayes	Count	.6439	.6523	.6541	.6569	.6309	.6327	.6414	.6439	.6471	.6184
	TF-IDF	.6503	.6593	.6652	.6667	.6339	.6345	.6425	.6589	.6569	.6121
Passive Aggressive	Count	.5912	.5975	.5971	.5980	.5844	.5912	.5975	.5971	.5980	.5844
	TF-IDF	.5889	.5974	.5970	.5980	.5799	.6138	.6154	.6199	.6139	.6138
Random Forests	Count	.6310	.6368	.6365	.6373	.6247	.6448	.6505	.6505	.6505	.6392
	TF-IDF	.6575	.6656	.6651	.6667	.6484	.6573	.6646	.6645	.6667	.6480
XG Boost	Count	.5866	.5877	.6256	.6214	.5518	.5942	.5921	.6391	.6275	.5609
	TF-IDF	.5927	.5899	.6475	.6311	.5544	.5918	.5864	.6452	.6275	.5562

TABLE 24. Results of CR-based sentiment classification on dataset D_{BASE}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.6873	.6816	.6937	.6832	.6914	.6431	.6213	.6691	.6337	.6525
	TF-IDF	.6352	.6337	.6368	.6337	.6368	.6389	.6374	.6407	.6373	.6405
Linear SVC	Count	.6978	.6989	.6988	.6990	.6967	.7702	.7717	.7726	.7723	.7682
	TF-IDF	.7526	.7538	.7564	.7549	.7503	.7715	.7728	.7786	.7745	.7684
Logistic Regression Class.	Count	.7627	.7643	.7646	.7647	.7608	.7797	.7814	.7831	.7822	.7772
	TF-IDF	.7636	.7655	.7694	.7670	.7602	.7465	.7481	.7625	.7525	.7404
Naïve Bayes	Count	.7390	.7409	.7445	.7426	.7354	.7280	.7255	.7305	.7255	.7305
	TF-IDF	.6979	.6996	.7139	.7059	.6900	.6944	.6930	.7294	.7059	.6830
Passive Aggressive	Count	.6748	.6763	.6762	.6765	.6731	.7720	.7739	.7747	.7745	.7695
	TF-IDF	.6952	.6959	.6959	.6961	.6944	.7797	.7814	.7831	.7822	.7772
Random Forests	Count	.7720	.7739	.7747	.7745	.7695	.7722	.7735	.7763	.7745	.7700
	TF-IDF	.7720	.7739	.7747	.7745	.7695	.7697	.7725	.7780	.7745	.7648
XG Boost	Count	.7722	.7735	.7763	.7745	.7700	.7797	.7814	.7831	.7822	.7772
	TF-IDF	.7797	.7814	.7831	.7822	.7772	.7797	.7814	.7831	.7822	.7772

TABLE 25. Results of CR-based sentiment classification on dataset D_{TAGS}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.6632	.6538	.6728	.6569	.6695	.6157	.5924	.6394	.6078	.6236
	TF-IDF	.6371	.6374	.6376	.6373	.6369	.6644	.6662	.6661	.6667	.6620
Linear SVC	Count	.6960	.6962	.6964	.6961	.6959	.7720	.7739	.7747	.7745	.7695
	TF-IDF	.7432	.7443	.7458	.7451	.7412	.7697	.7725	.7780	.7745	.7648
Logistic Regression Class.	Count	.7534	.7544	.7552	.7549	.7518	.7797	.7814	.7831	.7822	.7772
	TF-IDF	.7787	.7808	.7850	.7822	.7752	.7617	.7636	.7754	.7670	.7564
Naïve Bayes	Count	.7441	.7460	.7496	.7476	.7407	.7164	.7158	.7171	.7157	.7171
	TF-IDF	.7126	.7137	.7260	.7184	.7067	.7181	.7145	.7542	.7255	.7108
Passive Aggressive	Count	.6667	.6669	.6677	.6667	.6667	.7720	.7739	.7747	.7745	.7695
	TF-IDF	.6701	.6701	.6708	.6699	.6702	.7729	.7749	.7805	.7767	.7690
Random Forests	Count	.7797	.7814	.7831	.7822	.7772	.7801	.7828	.7868	.7843	.7759
	TF-IDF	.7797	.7814	.7831	.7822	.7772	.7719	.7740	.7836	.7767	.7671
XG Boost	Count	.7738	.7756	.7783	.7767	.7709	.7738	.7756	.7783	.7767	.7709
	TF-IDF	.7813	.7835	.7851	.7843	.7782	.7738	.7756	.7783	.7767	.7709

TABLE 26. Results of CR-based sentiment classification on dataset D_{TEXT}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.6675	.6617	.6733	.6634	.6715	.6840	.6722	.6986	.6765	.6916
	TF-IDF	.6470	.6473	.6481	.6471	.6469	.6417	.6431	.6431	.6436	.6398
Linear SVC	Count	.7153	.7158	.7161	.7157	.7149	.7720	.7739	.7747	.7745	.7695
	TF-IDF	.7400	.7417	.7431	.7426	.7374	.7613	.7625	.7700	.7647	.7578
Logistic Regression Class.	Count	.7534	.7544	.7552	.7549	.7518	.7722	.7735	.7763	.7745	.7700
	TF-IDF	.7683	.7704	.7761	.7723	.7643	.7515	.7532	.7672	.7573	.7457
Naïve Bayes	Count	.7211	.7236	.7268	.7255	.7166	.7164	.7159	.7174	.7157	.7172
	TF-IDF	.6963	.6969	.7117	.7030	.6897	.6934	.6903	.7276	.7030	.6838
Passive Aggressive	Count	.6771	.6768	.6782	.6765	.6777	.7228	.7239	.7274	.7255	.7200
	TF-IDF	.6852	.6863	.6863	.6863	.6841	.7339	.7358	.7406	.7379	.7300
Random Forests	Count	.7797	.7814	.7831	.7822	.7772	.7697	.7725	.7780	.7745	.7648
	TF-IDF	.7693	.7711	.7739	.7723	.7662	.7673	.7694	.7793	.7723	.7623
XG Boost	Count	.7693	.7711	.7739	.7723	.7662	.7693	.7711	.7739	.7723	.7662
	TF-IDF	.7616	.7638	.7653	.7647	.7584	.7693	.7711	.7739	.7723	.7662

TABLE 27. Results of CR-based sentiment classification on dataset D_{EMOJI}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.7512	.7536	.7560	.7549	.7474	.7523	.7543	.7549	.7549	.7497
	TF-IDF	.6898	.6920	.6992	.6961	.6836	.7303	.7329	.7377	.7353	.7254
Linear SVC	Count	.7720	.7739	.7747	.7745	.7695	.7439	.7448	.7451	.7451	.7428
	TF-IDF	.7553	.7567	.7575	.7573	.7533	.7358	.7373	.7379	.7379	.7338
Logistic Regression Class.	Count	.7720	.7739	.7747	.7745	.7695	.7807	.7819	.7822	.7822	.7791
	TF-IDF	.7720	.7739	.7747	.7745	.7695	.7738	.7756	.7783	.7767	.7709
Naïve Bayes	Count	.7693	.7711	.7739	.7723	.7662	.7616	.7638	.7653	.7647	.7584
	TF-IDF	.7118	.7124	.7207	.7157	.7079	.7484	.7504	.7557	.7525	.7444
Passive Aggressive	Count	.6468	.6483	.6506	.6505	.6432	.6169	.6178	.6181	.6176	.6161
	TF-IDF	.6829	.6851	.6860	.6863	.6795	.7049	.7059	.7059	.7059	.7038
Random Forests	Count	.7514	.7523	.7523	.7525	.7503	.7553	.7567	.7575	.7573	.7533
	TF-IDF	.7712	.7721	.7722	.7723	.7702	.7720	.7739	.7747	.7745	.7695
XG Boost	Count	.7813	.7835	.7851	.7843	.7782	.7813	.7835	.7851	.7843	.7782
	TF-IDF	.7917	.7936	.7944	.7941	.7892	.7917	.7936	.7944	.7941	.7892

TABLE 28. Results of CR-based sarcasm classification on dataset D_{BASE}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.6789	.6484	.7132	.6569	.7010	.5313	.3816	.6218	.4851	.5775
	TF-IDF	.6450	.6376	.6530	.6373	.6528	.6579	.6580	.6623	.6569	.6589
Linear SVC	Count	.7041	.7065	.7076	.7059	.7023	.7614	.7647	.7647	.7647	.7581
	TF-IDF	.7484	.7522	.7520	.7525	.7444	.7567	.7632	.7641	.7647	.7487
Logistic Regression Class.	Count	.7415	.7451	.7451	.7451	.7379	.7796	.7822	.7822	.7822	.7769
	TF-IDF	.7567	.7632	.7641	.7647	.7487	.7353	.7445	.7638	.7525	.7182
Naïve Bayes	Count	.7187	.7255	.7279	.7282	.7093	.7117	.7096	.7152	.7087	.7146
	TF-IDF	.6784	.6858	.7303	.7059	.6508	.6488	.6433	.7496	.6832	.6145
Passive Aggressive	Count	.6749	.6792	.6790	.6796	.6701	.7614	.7647	.7647	.7647	.7581
	TF-IDF	.6620	.6667	.6667	.6667	.6573	.7702	.7742	.7741	.7745	.7658
Random Forests	Count	.7885	.7918	.7917	.7921	.7849	.7774	.7816	.7817	.7822	.7726
	TF-IDF	.7702	.7742	.7741	.7745	.7658	.7783	.7831	.7844	.7843	.7724
XG Boost	Count	.7813	.7843	.7843	.7843	.7783	.7796	.7822	.7822	.7822	.7769
	TF-IDF	.7813	.7843	.7843	.7843	.7783	.7813	.7843	.7843	.7843	.7783

TABLE 29. Results of CR-based sarcasm classification on dataset D_{TAGS}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.6374	.5868	.6906	.6078	.6670	.5157	.3316	.6179	.4653	.5660
	TF-IDF	.6515	.6481	.6572	.6471	.6559	.6667	.6677	.6708	.6667	.6667
Linear SVC	Count	.7018	.7059	.7059	.7059	.6976	.7591	.7641	.7640	.7647	.7534
	TF-IDF	.7477	.7531	.7550	.7549	.7404	.7632	.7715	.7761	.7745	.7518
Logistic Regression Class.	Count	.7480	.7538	.7541	.7549	.7410	.7789	.7837	.7837	.7843	.7736
	TF-IDF	.7619	.7694	.7742	.7723	.7515	.7339	.7446	.7712	.7549	.7129
Naïve Bayes	Count	.7520	.7609	.7671	.7647	.7394	.7127	.7136	.7155	.7129	.7125
	TF-IDF	.6871	.6945	.7474	.7157	.6586	.6377	.6292	.7421	.6733	.6021
Passive Aggressive	Count	.6861	.6918	.6917	.6931	.6792	.7591	.7641	.7640	.7647	.7534
	TF-IDF	.6819	.6863	.6863	.6863	.6775	.7641	.7704	.7725	.7723	.7559
Random Forests	Count	.7789	.7837	.7837	.7843	.7736	.7655	.7726	.7745	.7745	.7565
	TF-IDF	.7789	.7837	.7837	.7843	.7736	.7655	.7726	.7745	.7745	.7565
XG Boost	Count	.7702	.7742	.7741	.7745	.7658	.7731	.7764	.7763	.7767	.7694
	TF-IDF	.7702	.7742	.7741	.7745	.7658	.7702	.7742	.7741	.7745	.7658

TABLE 30. Results of CR-based sarcasm classification on dataset D_{TEXT}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.6854	.6615	.7110	.6667	.7041	.6006	.5172	.6875	.5644	.6368
	TF-IDF	.6649	.6572	.6730	.6569	.6730	.6748	.6742	.6785	.6733	.6763
Linear SVC	Count	.6930	.6964	.6969	.6961	.6899	.7725	.7748	.7751	.7745	.7705
	TF-IDF	.7392	.7444	.7443	.7451	.7333	.7668	.7740	.7787	.7767	.7569
Logistic Regression Class.	Count	.7395	.7426	.7426	.7426	.7364	.7813	.7843	.7843	.7843	.7783
	TF-IDF	.7730	.7799	.7834	.7822	.7639	.7140	.7242	.7488	.7353	.6928
Naïve Bayes	Count	.7322	.7410	.7465	.7451	.7192	.7175	.7166	.7210	.7157	.7194
	TF-IDF	.6561	.6609	.7131	.6863	.6260	.6292	.6243	.7183	.6667	.5918
Passive Aggressive	Count	.6643	.6673	.6685	.6667	.6620	.7614	.7647	.7647	.7647	.7581
	TF-IDF	.6794	.6832	.6832	.6832	.6756	.7567	.7632	.7641	.7647	.7487
Random Forests	Count	.7901	.7939	.7937	.7941	.7860	.7752	.7809	.7821	.7822	.7682
	TF-IDF	.7685	.7720	.7719	.7723	.7646	.7520	.7609	.7671	.7647	.7394
XG Boost	Count	.7813	.7843	.7843	.7843	.7783	.7813	.7843	.7843	.7843	.7783
	TF-IDF	.7796	.7822	.7822	.7822	.7769	.7813	.7843	.7843	.7843	.7783

TABLE 31. Results of CR-based sarcasm classification on dataset D_{EMOJI}.

Classifier	Vectorizer	Uni-gram					n-gram				
		BA	F	P	R	S	BA	F	P	R	S
k-NN	Count	.7836	.7847	.7858	.7843	.7829	.7811	.7827	.7837	.7822	.7800
	TF-IDF	.7304	.7350	.7348	.7353	.7255	.7349	.7382	.7386	.7379	.7319
Linear SVC	Count	.7725	.7748	.7751	.7745	.7705	.7470	.7521	.7519	.7525	.7415
	TF-IDF	.7774	.7836	.7836	.7843	.7705	.7567	.7632	.7641	.7647	.7487
Logistic Regression Class.	Count	.7915	.7944	.7948	.7941	.7890	.7759	.7815	.7815	.7822	.7697
	TF-IDF	.7877	.7932	.7937	.7941	.7813	.7942	.8018	.8055	.8039	.7844
Naïve Bayes	Count	.7591	.7641	.7640	.7647	.7534	.7591	.7641	.7640	.7647	.7534
	TF-IDF	.7065	.7188	.7269	.7255	.6875	.7497	.7595	.7700	.7647	.7347
Passive Aggressive	Count	.7368	.7435	.7442	.7451	.7286	.7004	.7002	.7046	.6990	.7017
	TF-IDF	.7410	.7468	.7467	.7476	.7344	.7193	.7247	.7245	.7255	.7131
Random Forests	Count	.7503	.7546	.7544	.7549	.7457	.7503	.7546	.7544	.7549	.7457
	TF-IDF	.7678	.7735	.7739	.7745	.7612	.7789	.7837	.7837	.7843	.7736
XG Boost	Count	.8012	.8039	.8039	.8039	.7984	.8012	.8039	.8039	.8039	.7984
	TF-IDF	.8074	.8116	.8115	.8119	.8030	.8027	.8058	.8058	.8058	.7995

DATA AVAILABILITY

The dataset can be requested by emailing the corresponding author of this article.

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this article.

ACKNOWLEDGMENT

The authors would like to thank the people who worked with them in preparing the dataset: Mirza Muhammad Muneeb Baig, Tooba Arshad, Abdul Basit Akazai, Iqra Fahad, Kanwal Jabeen, and Maira Shahwar; without these superheroes, this work is not possible. They also maintain that a little portion of text in this article is rephrased using the ChatGPT, for seeking academic simplification. However, the original/initial text given to the ChatGPT was purely autogenous for this article, and it does not overlap with any other content.

REFERENCES

[1] S. M. Mohammad, "Ethics sheet for automatic emotion recognition and sentiment analysis," *Comput. Linguistics*, vol. 48, no. 2, pp. 239–278, 2022.

[2] C. E. Izard, "Basic emotions, relations among emotions, and emotion-cognition relations," *Psychol. Rev.*, vol. 99, no. 3, pp. 561–565, 1992.

[3] M. A. Hogg and D. Abrams, "Social cognition and attitudes," in *Psychology*, G. N. Martin, N. R. Carlson, and W. Buskist, Eds., 3rd ed. Pearson Education Limited, 2007, pp. 684–721.

[4] E. Harmon-Jones, C. Harmon-Jones, D. M. Amodio, and P. A. Gable, "Attitudes toward emotions," *J. Personality Social Psychol.*, vol. 101, no. 6, pp. 1332–1350, 2011.

[5] C. D. Batson, L. L. Shaw, and K. C. Oleson, "Differentiating affect, mood, and emotion: Toward functionally based conceptual distinctions," in *Emotion*. Newbury Park, CA, USA: Sage, 1992.

[6] E. A. Blechman, *Moods, Affect, and Emotions*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1990.

[7] S. M. Mohammad, "Sentiment analysis: Detecting valence, emotions, and other affectual states from text," in *Emotion Measurement*. Amsterdam, The Netherlands: Elsevier, 2016, pp. 201–237.

[8] Z. Nasim and S. Ghani, "Sentiment analysis on Urdu tweets using Markov chains," *Social Netw. Comput. Sci.*, vol. 1, no. 5, pp. 1–13, Sep. 2020.

[9] I. Safder, Z. Mahmood, R. Sarwar, S.-U. Hassan, F. Zaman, R. M. A. Nawab, F. Bukhari, R. A. Abbasi, S. Alelyani, N. R. Aljohani, and R. Nawaz, "Sentiment analysis for Urdu online reviews using deep learning models," *Expert Syst.*, vol. 38, no. 8, Dec. 2021, Art. no. e12751.

[10] M. S. Nizami, M. Y. Khan, and T. Ahmed, "Towards a generic approach for PoS-tagwise lexical similarity of languages," in *Proc. Int. Conf. Intell. Technol. Appl.* Singapore: Springer, 2019, pp. 493–501.

- [11] A. Julka, "From Hindi to Urdu: A social and political history," *Strategic Anal.*, vol. 36, no. 5, pp. 829–830, Sep. 2012.
- [12] M. S. Nizami, T. Ahmed, and M. Yaseen, "Hindustani or Hindi vs. Urdu: A computational approach for the exploration of similarities under phonetic aspects," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, pp. 749–755, 2020.
- [13] T. Ahmed, M. Suffian, M. Y. Khan, and A. Bogliolo, "Discovering lexical similarity using articulatory feature-based phonetic edit distance," *IEEE Access*, vol. 10, pp. 1533–1544, 2022.
- [14] M. M. Bradley and P. J. Lang, "Affective Norms for English Words (ANEW): Instruction manual and affective ratings," Center Res. Psychophysiol., Univ. Florida, Gainesville, FL, USA, Tech. Rep. C-1, 1999, vol. 30, no. 1.
- [15] E. Cambria, D. Olsher, and D. Rajagopal, "SenticNet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1515–1521.
- [16] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," 2011, *arXiv:1103.2903*.
- [17] S. M. Mohammad and P. D. Turney, "NRC emotion lexicon," Nat. Res. Council, Canada, 2013, p. 234, vol. 2.
- [18] M. Y. Khan, S. M. Emaduddin, and K. N. Junejo, "Harnessing English sentiment lexicons for polarity detection in Urdu tweets: A baseline approach," in *Proc. IEEE 11th Int. Conf. Semantic Comput. (ICSC)*, Jan. 2017, pp. 242–249.
- [19] M. Y. Khan and M. S. Nizami, "Urdu Sentiment Corpus (v1.0): Linguistic exploration and visualization of labeled dataset for Urdu sentiment analysis," in *Proc. Int. Conf. Inf. Sci. Commun. Technol. (ICISCT)*, Feb. 2020, pp. 1–15.
- [20] J. Berkson, "Application of the logistic function to bio-assay," *J. Amer. Stat. Assoc.*, vol. 39, no. 227, pp. 357–365, Sep. 1944.
- [21] J. S. Cramer, "The origins of logistic regression," Tinbergen Inst., Work. Paper 2002-119/4, Dec. 2002. [Online]. Available: <https://ssrn.com/abstract=360300>
- [22] N. Mukhtar, M. A. Khan, N. Chiragh, and S. Nazir, "Identification and handling of intensifiers for enhancing accuracy of Urdu sentiment analysis," *Expert Syst.*, vol. 35, no. 6, Dec. 2018, Art. no. e12317.
- [23] N. Mukhtar, M. A. Khan, N. Chiragh, A. U. Jan, and S. Nazir, "Recognition and effective handling of negations in enhancing the accuracy of Urdu sentiment analyzer," *Mehran Univ. Res. J. Eng. Technol.*, vol. 39, no. 4, pp. 759–771, Oct. 2020.
- [24] M. Hassan and M. Shoaib, "Opinion within opinion: Segmentation approach for Urdu sentiment analysis," *Int. Arab J. Inf. Technol.*, vol. 15, no. 1, pp. 21–28, 2018.
- [25] R. Bibi, U. Qamar, M. Ansar, and A. Shaheen, "Sentiment analysis for Urdu news tweets using decision tree," in *Proc. IEEE 17th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, May 2019, pp. 66–70.
- [26] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.
- [27] U. Sehar, S. Kanwal, K. Dashtipur, U. Mir, U. Abbasi, and F. Khan, "Urdu sentiment analysis via multimodal data mining based on deep learning algorithms," *IEEE Access*, vol. 9, pp. 153072–153082, 2021.
- [28] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
- [29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [30] U. Naqvi, A. Majid, and S. A. Abbas, "UTSA: Urdu text sentiment analysis using deep learning methods," *IEEE Access*, vol. 9, pp. 114085–114094, 2021.
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.
- [32] L. Khan, A. Amjad, N. Ashraf, H.-T. Chang, and A. Gelbukh, "Urdu sentiment analysis with deep learning methods," *IEEE Access*, vol. 9, pp. 97803–97812, 2021.
- [33] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 8, 1995, pp. 1–7.
- [34] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Int. Conf. Mach. Learn.*, vol. 96. Princeton, NJ, USA: Citeseer, Jul. 1996, pp. 148–156.
- [35] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *J. Jpn. Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771–780, 1999.
- [36] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Jun. 2016.
- [37] 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, pp. 1–17, Mar. 2022.
- [38] M. Y. Khan and K. Nazir, "Exerting 2D-space of sentiment lexicons with machine learning techniques: A hybrid approach for sentiment analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 6, pp. 599–608, 2020.
- [39] M. S. M. Suhaimin, M. H. A. Hijazi, R. Alfred, and F. Coenen, "Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts," in *Proc. 8th Int. Conf. Inf. Technol. (ICIT)*, May 2017, pp. 703–709.
- [40] S. Mukherjee, "Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering," *Technol. Soc.*, vol. 48, pp. 19–27, Feb. 2017.
- [41] S. K. Bharti, K. S. Babu, and R. Raman, "Context-based sarcasm detection in Hindi tweets," in *Proc. 9th Int. Conf. Adv. Pattern Recognit. (ICAPR)*, Dec. 2017, pp. 1–6.
- [42] S. K. Bharti, K. S. Babu, and S. K. Jena, "Harnessing online news for sarcasm detection in Hindi tweets," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell. Cham, Switzerland: Springer*, 2017, pp. 679–686.
- [43] S. Swami, A. Khandelwal, V. Singh, S. S. Akhtar, and M. Shrivastava, "A corpus of English-Hindi code-mixed tweets for sarcasm detection," 2018, *arXiv:1805.11869*.
- [44] M. J. C. Samonte, C. J. T. Dollete, P. M. M. Capanas, M. L. C. Flores, and C. B. Soriano, "Sentence-level sarcasm detection in English and Filipino tweets," in *Proc. 4th Int. Conf. Ind. Bus. Eng.*, 2018, pp. 181–186.
- [45] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, "CASCADE: Contextual sarcasm detection in online discussion forums," 2018, *arXiv:1805.06413*.
- [46] S. Amir, B. C. Wallace, H. Lyu, and P. C. M. J. Silva, "Modelling context with user embeddings for sarcasm detection in social media," *arXiv:1607.00976*.
- [47] Y. Ren, D. Ji, and H. Ren, "Context-augmented convolutional neural networks for Twitter sarcasm detection," *Neurocomputing*, vol. 308, pp. 1–7, Sep. 2018.
- [48] D. D. Thimmappa, "Paragraph vector based sarcasm detection in text," Ph.D. dissertation, School Comput., Nat. College Ireland, Dublin, Republic of Ireland, 2019.
- [49] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2013, pp. 746–751.
- [50] Y. Cai, H. Cai, and X. Wan, "Multi-modal sarcasm detection in Twitter with hierarchical fusion model," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2506–2515.
- [51] M. Bedi, S. Kumar, M. S. Akhtar, and T. Chakraborty, "Multi-modal sarcasm detection and humor classification in code-mixed conversations," *IEEE Trans. Affect. Comput.*, vol. 14, no. 2, pp. 1363–1365, Apr./Jun. 2021.
- [52] D. Jain, A. Kumar, and G. Garg, "Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106198.
- [53] A. Muhammad, N. Wiratunga, and R. Lothian, "Context-aware sentiment analysis of social media," in *Advances in Social Media Analysis*. Cham, Switzerland: Springer, 2015, pp. 87–104.
- [54] V. D. Bhat, V. S. Deshpande, and R. Sugandhi, "A multimodal sentiment analysis scheme to detect hidden sentiments," in *4th Post Graduate Conf. iPGCON*, 2014.

- [55] R. A. Potamias, G. Siolas, and A.-G. Stafylopatis, "A transformer-based approach to irony and sarcasm detection," *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17309–17320, Dec. 2020.
- [56] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
- [57] A. I. Alharbi and M. Lee, "Multi-task learning using a combination of contextualised and static word embeddings for Arabic sarcasm detection and sentiment analysis," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, pp. 318–322, 2021.
- [58] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," 2020, *arXiv:2101.01785*.
- [59] P. Kumar and G. Sarin, "WELMSD—Word embedding and language model based sarcasm detection," *Online Inf. Rev.*, vol. 46, no. 7, pp. 1242–1256, Oct. 2022.
- [60] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
- [61] R. Badlani, N. Asnani, and M. Rai, "Disambiguating sentiment: An ensemble of humour, sarcasm, and hate speech features for sentiment classification," in *Proc. W-NUT*, 2019, pp. 337–345.
- [62] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
- [63] M. Khodak, N. Saunshi, and K. Vodrahalli, "A large self-annotated corpus for sarcasm," 2017, *arXiv:1704.05579*.
- [64] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. Int. AAAI Conf. Web Social Media*, 2017, vol. 11, no. 1, pp. 512–515.
- [65] S. R. Munoz and S. I. Bangdiwala, "Interpretation of Kappa and B statistics measures of agreement," *J. Appl. Statist.*, vol. 24, no. 1, pp. 105–112, Feb. 1997.
- [66] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, p. 159, Mar. 1977.
- [67] N. Durrani and S. Hussain, "Urdu word segmentation," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Technol.*, 2010, pp. 528–536.
- [68] K. Pearson, "Mathematical contributions to the theory of evolution—XII. On a generalised theory of alternative inheritance, with special reference to Mendel's laws," *Proc. Roy. Soc. London*, vol. 72, nos. 477–486, pp. 505–509, 1904.
- [69] S.-Y. Lee, W.-Y. Poon, and P. M. Bentler, "A two-stage estimation of structural equation models with continuous and polytomous variables," *Brit. J. Math. Stat. Psychol.*, vol. 48, no. 2, pp. 339–358, Nov. 1995.
- [70] D. G. Bonett and R. M. Price, "Inferential methods for the tetrachoric correlation coefficient," *J. Educ. Behav. Statist.*, vol. 30, no. 2, pp. 213–225, Jun. 2005.
- [71] K. Pearson, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 50, no. 302, pp. 157–175, 1900.
- [72] G. U. Yule, "On the methods of measuring association between two attributes," *J. Roy. Stat. Soc.*, vol. 75, no. 6, pp. 579–652, May 1912.
- [73] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica Biophysica Acta (BBA) Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975.
- [74] J. P. Guilford, *Psychometric Methods*. New York, NY, USA: McGraw-Hill, 1954.
- [75] H. Cramér, *Mathematical Methods of Statistics*, vol. 43. Princeton, NJ, USA: Princeton Univ. Press, 1999.
- [76] K. Kafadar, "Handbook of parametric and nonparametric statistical procedures," *Amer. Statistician*, vol. 51, no. 4, pp. 374–375, 1997.
- [77] H.-F. Yu, F.-L. Huang, and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," *Mach. Learn.*, vol. 85, nos. 1–2, pp. 41–75, Oct. 2011.
- [78] W. Muhammad, M. Mushtaq, K. N. Junejo, and M. Y. Khan, "Sentiment analysis of product reviews in the absence of labelled data using supervised learning approaches," *Malaysian J. Comput. Sci.*, vol. 33, no. 2, pp. 118–132, Apr. 2020.
- [79] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to Information Retrieval*, vol. 39., Cambridge, U.K.: Cambridge Univ. Press, 2008.
- [80] Z. Harry, "The optimality of naive Bayes," in *Proc. Florida Artif. Intell. Res. Soc. Conf.* Washington, DC, USA: American Association for Artificial Intelligence, 2004, pp. 1–6.
- [81] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," 2013, *arXiv:1302.4964*.
- [82] H. Zhang, "Exploring conditions for the optimality of naive Bayes," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 19, no. 2, pp. 183–198, 2005.
- [83] M. Y. Khan, A. Qayoom, M. S. Nizami, M. S. Siddiqui, S. Wasi, and S. M. K.-U.-R. Raazi, "Automated prediction of Good Dictionary EXamples (GDEX): A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques," *Complexity*, vol. 2021, Sep. 2021, Art. no. 2553199.
- [84] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
- [85] G. Biau, "Analysis of a random forests model," *J. Mach. Learn. Res.*, vol. 13, pp. 1063–1095, Apr. 2012.
- [86] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.
- [87] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, Mar. 2006.
- [88] E. Fix and J. L. Hodges, "Discriminatory analysis. Nonparametric discrimination: Consistency properties," *Int. Stat. Rev./Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, Dec. 1989.
- [89] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, p. 175, Aug. 1992.
- [90] S. Shaikh, M. Y. Khan, and M. S. Nizami, "Using patient descriptions of 20 most common diseases in text classification for evidence-based medicine," in *Proc. Mohammad Ali Jinnah Univ. Int. Conf. Comput. (MAJICC)*, Jul. 2021, pp. 1–8.
- [91] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 3121–3124.
- [92] J. D. Kelleher, B. M. Namee, and A. D'arcy, *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. Cambridge, MA, USA: MIT Press, 2020.
- [93] I. Guyon, K. Bennett, G. Cawley, H. J. Escalante, S. Escalera, T. K. Ho, N. Macià, B. Ray, M. Saeed, A. Statnikov, and E. Viegas, "Design of the 2015 ChaLearn AutoML challenge," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.



MUHAMMAD YASEEN KHAN received the B.S. degree in software engineering from the University of Karachi, in 2010, and the M.S. degree in software engineering from the Karachi Institute of Economics and Technology, in 2015. He is currently pursuing the Ph.D. degree in computer sciences with Mohammad Ali Jinnah University, Pakistan. His professional background encompasses positions as a Software Development Engineer with Mazik Global, a Principal Research and Development Engineer with Love For Data, and a Senior Data Scientist with Daraz-Alibaba Group Inc. His current research interests include the integration of AI, psychology, and NLP to advance scientific knowledge in this domain.



TAFSEER AHMED received the Ph.D. degree from Universität Konstanz, Germany, in 2009. With over 20 years of combined teaching and research experience, his expertise spans various areas in natural language processing (NLP). Notably, he has contributed to lexical functional grammar, universal dependencies, PoS tagging, author attribution, transliteration, named entity recognition, and other NLP applications, particularly focusing on Urdu and Pakistani languages.

He notably participated in the development of Urdu Propbank with the University of Colorado Boulder and acted as the Co-PI of the DAAD Project “Urdu Text to Speech: Understanding Intonation.” Furthermore, he held positions as the Head of NLP, Love For Data, and a Staff Scientist with QLU.ai.



SHAUKAT WASI received the B.S. degree in computer science from the University of Karachi and the M.S. and Ph.D. degrees in computer science from the FAST-National University of Computer and Emerging Sciences (NUCES). He was the founding Faculty Member of the Computer Science Department, DHA Suffa University, Karachi. Currently, he is an Associate Professor with the Faculty of Computing (FOC), Mohammad Ali Jinnah University (MAJU), Karachi,

where he leads the Interactive and Intelligent Natural Language Processing (IINLP) Research Group. His current research interests include text classification and mining, information retrieval and extraction, and human-computer interaction.

...



MUHAMMAD SHOAIB SIDDIQUI (Member, IEEE) received the B.S. degree from the Department of Computer Sciences, University of Karachi, in 2004, and the M.S. and Ph.D. degrees in computer engineering from Kyung Hee University, South Korea, in 2008 and 2012, respectively. Currently, he is an Associate Professor with the Islamic University of Madinah, Saudi Arabia. His current research interests include routing, security, and management in

wireless networks, sensor networks, IP traceback, secure provenance, blockchain technologies, and remote monitoring using the IoT. He is an esteemed member of ACM.