

Urdu Sentiment Analysis

Iffraah Rehman^{1*}, Tariq Rahim Soomro²

^{1,2} *CCSIS, Institute of Business Management (IoBM), Karachi, Pakistan*

Abstract – The world is heading towards more modernized and digitalized data and therefore a significant growth is observed in the active number of social media users with each passing day. Each post and comment can give an insight into valuable information about a certain topic or issue, a product or a brand, etc. Similarly, the process to uncover the underlying information from the opinion that a person keeps about any entity is called a sentiment analysis. The analysis can be carried out through two main approaches, i.e., either lexicon-based or machine learning algorithms. A significant amount of work in the different domains has been done in numerous languages for sentiment analysis, but minimal research has been conducted on the national language of Pakistan, which is Urdu. Twitter users who are familiar with Urdu update the tweets in two different textual formats either in Urdu Script (Nastaleeq) or in Roman Urdu. Thus, the paper is an attempt to perform the sentiment analysis on the Urdu language by extracting the tweets (Nastaleeq and Roman Urdu both) from Twitter using Tweepy API. A machine learning-based approach has been adopted for this study and the tool opted for the purpose is WEKA. The best algorithm was identified based on evaluation metrics, which comprise the number of correctly and incorrectly classified instances, accuracy, precision, and recall. SMO was found to be the most suitable machine learning algorithm for performing the sentiment analysis on Urdu (Nastaleeq) tweets, while the Roman Urdu Random Forest algorithm was identified as the best one.

Keywords – Machine learning algorithms, sentiment analysis, Tweepy, WEKA.

I. INTRODUCTION

Sentiment analysis can be defined to identify humans' views and emotional behaviour in textual form, towards any product, organisation, service, or individual. It is also a kind of data mining, which involves the extraction of human opinions through natural language processing (NLP) and text analysis. Sentiment analysis is often termed "Opinion Mining", for which the source of data extraction is usually the Web [1]. In this era, there are opinionated data of humans stored in digital form, which are received from different discussion forums, reviews, blogs, and available social media sites. Individuals seek the opinions of others, while making any decision; this is not only restricted to a person, but also organisations plan their marketing strategies based on the reviews of their customers. Thus, these opinions of others/customers are not limited to friends and family, but there are many public forums available on the Internet [2]. Sentiment analysis involves the categorization of positive, negative, and at times neutral facts

from the textual data available at hand. It is affecting a wide range of domains and applications such as enhancing sales and improving the marketing of a brand/product, identification of political strategies and analysing the trends, and forecasting the movement of the stock market by world news or financial reports available online [3]. Over time, the concept of sentiment analysis has grown beyond the detection of polarity; different techniques are now available, which can detect the emotions of the humans such as anger, happiness, and sadness. There are two techniques available through which the sentiment analysis can be performed. The first technique is machine learning-based, which uses the well-labelled training dataset, and then an automatic classifier is applied to it. The performance of the classifier is evaluated by how well it predicts the polarity of a sentiment(s) that is supplied to it through a test dataset. Machine learning can further be divided into supervised and unsupervised learning [4]. The second approach that can be adopted for the sentiment analysis is lexicon-based, which can assign the polarity to the sentence of a document based on the words available in its dictionary. In this case, a dictionary is a collection of positive and negative words that are either collected manually or can be extracted from already shared resources, such as SentiWordNet [5].

Apart from the English language, the sentiment analysis has also been performed for the Arabic language. The Arabic language is written from right to left and involves diacritical marks. It has 28 letters, 25 consonants, and 3 vowels. Several works have been done on the Arabic dataset, in which machine learning algorithms, i.e., SVM and Naïve Bayes have been applied to the dataset of Arabic tweets. An attempt has also been made to perform an aspect-based sentiment analysis, opinion holder extraction, and opinion spam detection [6]. One of the approaches of sentimental analysis lies in sentimental lexicon that can be either a word or a phrase, which conveys either a strong positive or strong negative emotion towards something be it any product or an event. Similar study has been performed in the Arabic language, in which a tweet regarding one brand has been extracted. Arabic sentiment ontology has been made consisting of 24 words that would express sentiments. Hence, positive and negative emotions have been evaluated based on ASO in that dataset of Arabic tweets [7].

Social media is an Internet-based platform, which is central to user-generated content and provides public membership. It is the most convenient way nowadays to propagate information to

* Corresponding author's e-mail: std_23371@iobm.edu.pk

others in the blink of an eye. There are several social network sites available, which are widely used by people around the globe among which the most used one are Facebook, Twitter, Instagram, YouTube, LinkedIn, etc. These sites can be categorized into dimensions such as conversations, information openness, connectivity, community formation, and participation [8]. Social media sites generate unstructured data in bulk amounts, on which the sentiment analysis can be applied to gain valuable insights. The data of all public posts posted on social media sites can be extracted through their respective APIs (Application Program Interface), such as Twitter has its user timeline API, streaming API, and REST API [9]. Similarly, Facebook is another largest platform for people for expressing their thoughts and ideas about areas such as politics, business, government, health, etc. These data in textual format can be extracted by using its Graph API [10]. Twitter was launched on 15 July 2006. It was first given the name *Twtrr*; however, soon after six months of its launch, its name was changed to *Twitter*. The adoption of Twitter among the users of social media started to increase in March 2007 as it was a micro-blogging site, which caught the attention of the users [11]. Similarly, since the first quarter of the year 2017, rapid growth in the number of Twitter active users has been observed [12], as shown in Fig. 1.

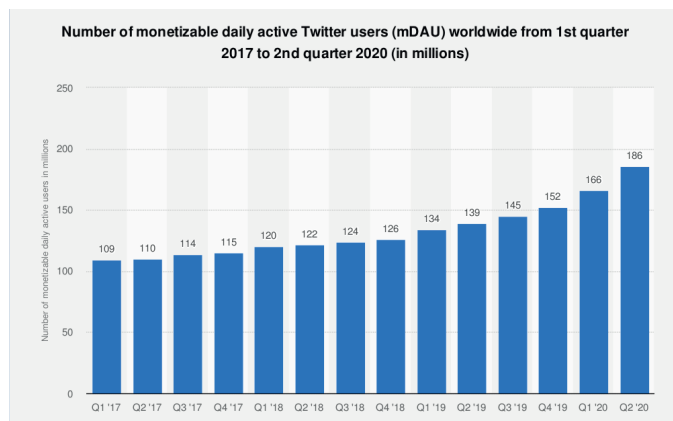


Fig. 1. Twitter statistics for active number of users from Q1 2017 to Q2 2020.

The time spent by users on Twitter is 3.39 minutes per session on average, on a daily basis. Moreover, the number of tweets sent each day equates to 5787 tweets per second [13]. Initially, Twitter could not tweet with a maximum of 140 characters only which allowed its users to be brief and creative with a post. It is termed a micro-blogging site; it allows the users to post and share the posts, which are termed tweets [14]. Currently, Twitter supports 34 languages, the English language being the topmost used language [15]. Urdu is the national language of Pakistan. It is an Indo-Aryan language having its roots in the Persian and Arabic languages. It is bidirectional language, i.e., the text is written from right to left, while the numbers are written from left to right. Urdu set consists of 38 alphabets; it has 7 diacritics marks which are optional, and 7 punctuation marks [16].

The main aim of this study is to perform the sentiment analysis specifically on the Urdu language (both Urdu and Roman Urdu). As far as social media platforms are concerned,

its users who can speak, understand, and write in Urdu update their status or writing comments in both Urdu scripts, i.e., Nastaleeq, and in Roman Urdu. Hence, this study is an attempt to perform the sentiment analysis on both Nastaleeq and Roman Urdu by extracting the data from Twitter using its API called Tweepy. This study will have its focus on experimenting with the sentiment analysis for the Urdu language by using one of its approaches, i.e., machine learning. Twitter will also be used to gather the data for Urdu and Roman Urdu tweets. A lot of work has been done on the sentiment analysis in the English language and fewer studies have been performed on the sentiment analysis in the Arabic language. However, detailed research has not been conducted to date for performing the sentiment analysis on Urdu (Nastaleeq) and Roman Urdu. There is certain limitation that is faced while performing the sentiment analysis on the Urdu language, such as a lack of lexicon, stop words, etc. Considering the social media and its users in Pakistan, they update their status on social media apps or write comments in Roman Urdu (using the English language to write in Urdu), as well as in actual Urdu (Nastaleeq), i.e., writing from right to left in the Urdu language fonts. There is no denying the fact that handling the Urdu script is an uphill task using the existing tools. The present paper is organised as follows. Section II covers a literature review to see what work has been done for the Urdu language; Section III discusses the material and methods used in this study; Section IV highlights the results and findings and Section V discusses the study along with future recommendations.

II. LITERATURE REVIEW

Content over social media has transformed over the years; it has become a rich source for getting people's opinions and views regarding the specific topic, brands, products, social issues, etc. Similarly, new research and techniques are emerging with the passing days of performing text mining and semantic analysis. This gives a path to gain valuable insights, which exist within the data obtained from the data being generated from social networking sites in all aspects of life whether it is healthcare, politics, etc. In [17], an attempt has been made by the author to analyse and predict the stock movement of Microsoft by extracting the data from Twitter and applying sentiment analysis to it. The data consisting of tweets comprising of keywords #MSFT, #Microsoft, #Windows, and other Microsoft brands have been collected for a certain date range of the year 2016. Also, the stock prices (opening and closing) of Microsoft have been obtained from yahoo. The data are first pre-processed through three steps such as tokenization, stop words removal, and regex. Extracted tweets are further categorized as positive, negative, and neutral. Two methods such as n-grams and word2vec have been used for textual representation. This process further passes through model training and is analysed for correlation of price and sentiment. Thus, it has been found that strong relationships exist between people's reviews about a company, expressed through tweets on Twitter, and the rise or fall of a company's stock price.

The social media platform "Twitter" is also widely used by sports fans; hence, the winner can already be predicted through

user thoughts and reviews posted on Twitter. Similarly, in [18], the winner of a premier league of football has been predicted. The real-time based data from Twitter has been extracted from Twitter API, which consists of all the tweets having hashtags related to the league and is stored in the Central Sport system, which is a tool for the Twitter collector and sentiment analysis. Match results are obtained from ESPN.com. Every tweet is analysed on two axes, such as tone and opinion finder. Separate models are built based on objectively classified tweets, all positive and all negative tweets. Social media being an open platform for all of its users around the globe give people freedom of speech. Whether it is the people protesting against some issue or supporting any good cause; all of the thoughts of people can be viewed and analysed on social media. Back in 2015, there were refugee crises in Europe and different views were coming from different people. The sentiment analysis on this crisis has also been performed in [19]. For this purpose, the tweets in two different languages, i.e., English and German, have been gathered. These data after being pre-processed and filtrated undergo a natural language processing tool called Linguistic Inquiry and Word Count (LIWC), which further filters the tweets into emotions like positive, negative, anger, and anxiety. The sentiment score has been calculated for each category and in each language. The dataset obtained from Twitter is mostly used for viewpoint analysis of users of different things. The views can be positive, neutral, or negative. One of the many ways to perform this analysis is to use a cluster-based method, such as in [20]. A new method CSK has been proposed by combining k-mean and cuckoo search. This method inputs the Twitter data and clusters them into three phases: (1) pre-processing of data, which removes all the noise from the data such as URLs, and hash (#) characters; (2) feature extraction, i.e., alphabets are removed according to the dictionary of stopwatch and acronym, and (3) hybrid clustering using k-means. Then, the tweets are converted into the feature vectors by segregating them into positive, negative, and neutral emoji, positive and negative exclamation and negation, and positive, negative, and neutral words. The results obtained from the proposed method are then compared with cuckoo search, improved cuckoo search, SVM-tri, and Naïve Bayes-tri based on statistical comparison of accuracy, computational time, and fitness functional value. Thus, the proposed method has been found most effective for the viewpoint analysis of Twitter data.

Twitter not having any language barrier, comprises millions of tweets every single day. These tweets are not only restricted to the English language, but users update their Twitter accounts in their native languages as well. Similarly, thousands of users are found that tweet in the Arabic language. Several attempts have already been made by different authors/researchers to perform the sentiment analysis in Arabic. However, Arabic is a very challenging language for the sentiment analysis, as it is written from right to left and may or may not have special characters, which can also be termed as diacritical marks. A hybrid approach has been proposed in [21], and manually labelling data for the machine learning classifier SVM has been tried. Twitter data have been collected using the Tweet Archivist tool, based on the hashtag of the social issue in Saudi

Arabia. Of 500 000 crawled tweets collected, only 1103 tweets have been left after data pre-processing and filtration. These data are passed through a lexicon-based classifier, which gives an output of labelled data, which is further passed through SVM to build a classifier model using the WEKA tool. These labelled classes are based on the negative and positive polarity of the tweets. The results are based on the accuracy and the confusion matrix. This study has proven that neither of the individual approaches such as lexicon-based classifier or SVM can achieve the highest accuracy of 84.01 % as this hybrid approach.

An attempt has been made in [22] to identify which machine learning algorithm best classifies the tweets either positive or negative regarding healthcare. The data from Twitter was extracted through Twitter API, using the keywords related to healthcare in Arabic. Initially, 126 959 tweets were extracted based on the keywords, such as *مستشفى تغلق الصحة* (Closing Hospital), *الصحة يعالج من* (Solving Health), *الصحة تحسن تنتظر* (Improving Health), *الصحى بالخدمات رأيك* (Opinions about Health). Then, the data were pre-processed and the noise was removed; the total number of tweets left was 2026. Further, the deep learning method was compared with the machine learning classification algorithms such as naïve Bayes, logistic regression, and SVM, based on calculated accuracy. Thus, SVM was found to be the best among all methods with accuracy between 85 % and 91 %. Apart from the application of linear models for the sentiment analysis on the Arabic data of tweets, now research has also been started for the application of deep learning models. Two models of deep learning such as Convolution Neural Network (CNN) and Long Short-Term Memory (LSTM) models have been combined to make an ensemble model [23]. This model then predicts the sentiment of Arabic tweets. All the tweets have a fixed size vector. CNN and LSTM were trained with different hyper-parameters. These data were divided into training (70 %), testing (10 %), and validation (20 %). Thus, the ensemble model gave an accuracy of 65.05 % and an F1 score of 64.46 %. Most of the work for the sentiment analysis has been done in the English language, while other languages hold minimal importance in the world of social media, such as Urdu, which is the national language of Pakistan. Twitter allows tweeting short blogs in Urdu as well. However, Urdu on Twitter can be found in two different ways, i.e., either in actual Urdu (Nastaliq) written from right to left (using the Urdu language fonts) or it can be written in Roman Urdu using the English language to write in Urdu. Very little research and practical work have been done on the sentiment analysis on Twitter in the Urdu language.

In [24] data based on 12 different hashtags were extracted. #CrimeMinisterNawaz, #JiyeBhutto, #MaryamMeriAwaz, and #PakStandsWithSC are to name a few. These tweets were first extracted in the Urdu language by Twitter API and then translated into Urdu through translator API. The actual dataset had 100 000 tweets, after pre-processing 6250 tweets were left. These tweets were first analysed through three different analysers, namely TextBlob, SentiWord, and W-WSD. TextBlob had the highest number of positive tweets, i.e., 3380 with 54.08 %, while SentiWord gave 3054 – the highest

negative number of tweets with 48.46 %. To verify the polarity obtained from these sentiment analysers, the data were further passed through two machine learning algorithms, i.e., Naïve Bayes and SVM. W-WSD analyser was found to give the highest number of correct instances along with 79.00 % for the given data using Naïve Bayes, while TextBlob gave an accuracy of 62.67 % using SVM. Another attempt was made in [25], in which sentiment analysis was performed on a bilingual dataset, i.e., data consisting of tweets in English as well as in Roman Urdu. The data were extracted based on the hashtags related to the General Election of Pakistan. 89 000 tweets were collected out of which 82 224 were further classified as political and 6847 as non-political. A bi-lingual sentiment lexicon (BSL) was developed using WordNet, SentiStrength, and a bilingual glossary. The sole purpose of BSL was to calculate the frequency of terms used in tweets in the language to give them strength also known as sentiment scores. Thus, English tweets had the strength of 2600 words and Roman Urdu had 3200 words. Results were analysed by evaluating the measure of recall, precision, accuracy, and F-measure.

Many tweets are updated in Urdu, which is related to different political situations prevailing in Pakistan; similarly, the polarity of these views can be measured by the sentiment analysis. An attempt was made in [26], in which data of 2703 tweets in Roman Urdu from Twitter were extracted based on the hashtags of cellular companies, such as Ufone, Zong, Mobilink, Warid, and Telenor by using Twitter API. The statistical calculations were done using the R language. The polarity of the tweet was analysed by giving a weighted score through SENTIWORDNET. Six machine learning algorithms such as bagging, SVM, Boosting, Forest Tree, Maxen, and Naïve Bayes were compared based on precision, recall, and F score. Thus, bagging was found to predict the best results of sentiments among all. The results of sentiment analysis may differ in different languages; in [27], Twitter data were extracted based on six different keywords in Urdu and English. The keywords were Pakistan, Imran Khan, Nawaz Sharif, Bhutto, justice, and Dam. First, the sentiment analysis was performed on English tweets of keywords and then on Urdu tweets. As for in English the keyword “Pakistan” had the highest strength of 42 %, while the keyword “Justice” had the lowest strength of 14 %. The keyword “Imran Khan” had the polarity of 9:7, i.e., 9 positive and 7 negative. However as of Urdu the keyword “پاکستان” had the highest strength of 20 % and “نواز شریف” had 0 % strength. Also, the keyword “Imran Khan” had a sentiment of 11:3 i.e., 11 positive and 3 negatives, unlike the result obtained in English for the same keyword which was 9:2. It was observed that Roman Urdu differs in many ways, in terms of how people write it. Everyone has a different style of spelling words. Thus, it does not have any standard for writing it. An attempt was made by the authors in [28] to perform the sentiment analysis on Roman Urdu. The data extracted was based on people’s reviews of different products on an e-commerce site known as Daraz.pk. The data of 80 285 reviews were collected and stored as vector space models, which were then labelled as negative, positive, and neutral. Later support vector machine algorithm was applied with the linear and cubic

kernel. To make SVM classify multiple classes, two approaches such as one-vs-one (OVO) and one-vs-all (OVA) were adopted. The algorithms were applied directly to complete data, and it was found that the cubic kernel was marginally better than the linear one as both of them showed the accuracy of sentiment N, i.e., Neutral as high, while sentiment analysis for E, i.e., negative remains a challenge. The technique of deep learning was also applied for the sentiment analysis in Roman Urdu. The author of the paper [29] used a deep neural network long short-term memory (LSTM) model on the Roman Urdu dataset. The training model was built by using four layers: the input (embedding) layer, the hidden layer, the LSTM layer, and the output layer. The cross-validation approach was applied with 10-folds in which 90 % was used as training while 10 % for the validation testing. The obtained results were compared with the machine learning-based classifiers that were Naïve Bayes, random forest, and SVM. Thus, it was found that the proposed deep learning model was able to give better results with the highest accuracy of 0.95180 among the other three machine learning classifiers. Similarly, in [30] it was found that around 64 million people spoke the Urdu language and knew about its Roman script. In the, a deep learning model was proposed which was able to mine the emotions and feelings of people in Roman Urdu. Two types of classifications were carried out known as binary classifications and tertiary classification. It was concluded that the RCNN model performed exceptionally better than the baseline models with an accuracy of 0.652 for binary classification and 0.572 for tertiary classification.

III. MATERIAL AND METHODS

This section of the research illustrates the methodology that has been adopted to perform the sentiment analysis of the Urdu language. There are two ways through which the sentiment analysis can be performed: one is lexicon-based, and the second is machine learning-based. Since the Urdu language is very rich in terms of morphology, creating lexicons in Urdu is a challenging task. Therefore, a machine learning approach has been selected for this study. First, the medium that is used for the data collection is Twitter, a well-known social media site. The tweets are extracted in the Urdu language, i.e., in Roman Urdu and the tweets are written in Urdu scripts (Nastaleeq). Access to Twitter data has been gained using one of twitter’s APIs called TWEETPY API. Second, the extracted data are pre-processed. Third, the pre-processed data are translated into the English language. Next, polarity is assigned to each tweet by using NLTK and TextBlob library of python. In the next step, the obtained data are prepared for training and testing and are further passed through several machine learning algorithms available in the WEKA tool.

A. Data Collection and Tools

Tweets that are updated in Urdu language (Nastaleeq and Roman) on Twitter have been extracted using Tweepy API. Python is used as a programming language for API and data extraction. Lastly, sublime text editor has been used as an editor for writing down the code.

B. Data Analysis Model

The extraction of tweets from Twitter begins by selecting the hashtags. The selected hashtags serve as the baseline for collecting the tweets related to one specific domain. In this study, several hashtags have been used that are related to the prevailing corona virus and its situation in Pakistan only.

#CoronaVirusPakistan	#stayhomestaysafe	#LockdownExtended
#CoronaInPakistan	#COVIDisnotFlu	#StayStrongImranKhan
#sealeddown	#PakistanFightsCorona	#PMIKFightsCorona
#COVID-19	#CoronaFreePakistan	#WeStandWithPMImranKhan
#COVID-19	#StaySafeStayHome	#TigerForce
#COVID-19	#StayAtHomeChallenge	#LockdownEnd
#LiftingRestrictions	#lockdownpakistan	#CoronaVirus
#PMIKCoronaFund	#PakistanUnitedAgainstCorona	#PunjabFightsCorona
#COVID19outbreak		

The extraction took place at different time intervals, depending upon the fact that either of the mentioned hashtags is trending on Twitter or not. Once a topic is trending on Twitter, the chances of getting a maximum number of tweets is getting higher. The Tweepy API supports the Urdu language; similarly, a python code was used along with the API to extract the tweets only in the Urdu language given the hashtags. A condition was added in the code to discard all the tweets that were in another language except for Urdu. During this extraction, the pre-processing of data was also done to save the clean tweets in a CSV file.

The total number of tweets gathered was 2300. In this dataset, there were some tweets, which only included unwanted text followed by a link such as “ویڈیو دیکھنے کے لیے نیچے لنک پر ”. Thus, such kinds of tweets were removed and the final dataset that was obtained had 1914 tweets, which were further used to perform the sentiment analysis. The pre-processing steps were performed.

The process to gather the tweets in Roman Urdu is similar to the extraction process of tweets in the Urdu language (Nastaleeq). Except for the fact that Roman Urdu is similar to the English language in terms of the alphabet and letters used for it. Similarly, the identification of tweets written in Roman Urdu that differs from actual English was a challenging task in the research. To overcome this challenge, the hashtags that were selected for this purpose were solely related to ongoing political issues in Pakistan. Also, a condition was added in the python code to exclude all the tweets that were found either in English or Urdu, leaving behind the tweets that were in Roman Urdu only based on the selected hashtags.

The hashtags used are provided below. 1127 is the total count of the tweets in Roman Urdu that were gathered after the extraction and basic pre-processing. The pre-processing steps were performed.

#CorruptionDushmanARY	#Patwari	#LanatARY
#calloffuniversitexams	#SindhDushmanPPP	#BilawalBhuttoTheLeader
#SindhDushmanARY	#supportARY	#ARYNewsUrdu
#ARYNews	#chorPPP	#ARYLanat
#PMIK	#BoycottARY	#WeWillExposeARY
#We_love_ARY	#iqrarulhasan	#PMImranKhan
#PPP	#GoNaiziGo	#ImranKhan
#extendlockdown	#TeamSareAam	#imrankhanPTI
#SaeedGhani	#BilawalBhuttoJawabDo	#ThankYouPMIK
#SindhGovt	#We_love_iqrar	#kaptan
#ARYNiazi	#bhuttozardari	#PEMRAShouldBanIqrarARY

The next step after the tweet extraction was to get the polarity of each tweet to get the labelled data. The labelled data were used to implement the machine learning approach of sentiment analysis. Python provided the packages, which could automatically detect the sentiment of a sentence whether the sentence was positive, negative, or neutral. These packages are NLTK (Natural Language Toolkit) and TextBlob. NLTK takes the string as an input, tokenizes it into words, removes the stop words, and then performs stemming. It also tags each word of a string into parts of speech. These words are then matched with the words available in the bag of words to get the sentiment scores. In NLTK, the bag of words that is used is called SentiWordNet. This is how NLTK works and assigns the polarity of each sentence in a dataset. TextBlob, in turn, is built upon NLTK, as it uses the NLTK corpora only. TextBlob identifies the polarity by calculating the number of positive and negative words in a sentence [31]. The mentioned two libraries of python do not provide support for the Urdu language. Also, there is not any python library, which has tokens for Urdu words, Urdu POS, complete and authentic list of positive and negative words. Due to this reason, the polarity of extracted Urdu tweets cannot be detected. To overcome this issue, the Urdu tweets were translated into English using the Google translator API provided by python. After the translation, the polarity was assigned using TextBlob and NLTK. A CSV file having Urdu tweets was passed through the python code with Google translator API. Once the tweets were translated into English, they were again saved into a CSV file. In the next step, the translated tweets were given the polarity by using NLTK and TextBlob. The tweets underwent some basic pre-processing again that was mentioned earlier to avoid unwanted noise that might have occurred during the translation. If the polarity of the tweet was greater than zero (0), then it was declared as positive; if it was less than zero (0), then it was declared as negative. Moreover, if the polarity of the tweet was exactly zero (0), then it was declared neutral. Table I depicts the polarity assignment to Roman Urdu Tweet as an example.

Roman Urdu is written using the English letters and alphabets, but it differs from the English language. Similarly, the python libraries NLTK and TextBlob cannot be used for gaining the polarity of Roman Urdu tweets. Therefore, in this study, the collected Roman Urdu tweets from Twitter were converted into the English language by using an online website <https://www.ijunoon.com/>. After the tweet conversion from Roman Urdu to English, the tweets were passed through NLTK and TextBlob to assign the polarity to each tweet. Table II depicts the polarity assignment to Urdu Tweet as an example.

Stop words can be defined as those words in a natural language that carry little or no information at all. These are the words that occur frequently in the documents, but they are less informative. Removal of such words from the text helps in significant information retrieval. The removal of stop words usually takes place in the pre-processing step of data. Thus, the elimination of such uninformative words from a sentence or document plays an important role in gaining better efficiency and accuracy of textual data [32]. Several works and studies were performed in the past for the development of stop word

list in the Urdu language for both Natalee and Roman Urdu. In [33], an algorithm was proposed, which removed the stop words from Urdu documents. The methodology of this algorithm was defined by using deterministic finite automaton (DFA). However, as a result, there was no stable list of stop words obtained by using the proposed algorithm. The authors of the paper [34] attempted to perform text classification in the Urdu language for which stop word removal was an essential step in pre-processing. Hence, a list of stop words was built manually based upon the data used. The list also included Arabic words. In this study, while considering the stop words for Urdu language (Nastaleeq), a list has been compiled for stop words from three different sources. The sources along with the number of words obtained from each source are mentioned below:

- *Ranks NL*: This is an online website that provides stop words in 40+ languages. A list consisting of 548 Urdu stop words has been obtained from this site. This source can be accessed at <https://www.ranks.nl/stopwords/urdu>. However, in the provided list some words have been found, which do not belong to the Urdu language, and some have been duplicated. Hence, after the elimination process, 286 stop words have been obtained from this source.
- The second source, which has been used to get the Urdu stop words, is the paper [35]. An algorithm was designed that identified the most frequently used words in each document. This algorithmic approach was implemented on the publicly available dataset of BBC Urdu news articles. The authors were able to get 93 different stop words for the Urdu language.
- The third source which has been used in this study is the paper [36]. The authors of the paper have attempted to present the list of 783 Urdu stop words. These words have been obtained by translating the available list of the English language stop words into Urdu.

Similarly, after the stop word data collection from the above-mentioned sources, the duplication was removed to get the unique set of words. The final list that has been obtained consists of 1003 stop words in the Urdu language (Nastaleeq script).

Urdu written in the Roman script is called Roman Urdu. Currently, there is no proper standard defined for writing Roman Urdu. It also has many irregularities in spelling; therefore, every other user on social media is updating their posts, tweets, or writing comments in their style in Roman Urdu. As far as this study is concerned, a list of stop words for Roman Urdu has been prepared manually. The stop words are extracted directly from the data used for the sentiment analysis, i.e., the tweets extracted in Roman Urdu from Twitter. The final list that has been obtained consists of 815 stop words of Roman Urdu.

C. WEKA

WEKA stands for Waikato Environment for Knowledge Analysis. This tool is easily available online and has a public license. WEKA supports several machine learning tasks, such

as classification, clustering, regression, attribute selection, visualization, and most importantly data pre-processing [37].

The testing and training dataset has been divided in the ratio of 70:30 (refer to Table III). The training set comprises 70 % of the tweets, while the remaining 30 % is used for testing. The total number of tweets in the Urdu language is 1914, out of which 1414 are reserved for the training purpose and the remaining 500 – for testing. As for Roman Urdu, the dataset consists of 1127 total tweets. Out of the total count, 789 tweets are used for training the machine learning algorithms, while 338 tweets are used for testing purposes.

The dataset is prepared in .arff file format that will be loaded into WEKA. The attributes of the dataset are the following:

```
@relation Urdu_SA;
@attribute Tweet_Urdu string;
@attribute 'Class' {'Positive','Negative','Neutral'}.
```

Pre-processing is the most important step that has to be performed before the application of any machine learning algorithm on the dataset. Similarly, in the research, the training is first loaded into WEKA using the explorer tab for pre-processing. Feature extraction is the first step in pre-processing. Feature extraction can be described as an initial reduction in a dataset without the loss of important and relevant information. It extracts the important features or variables from the dataset through different techniques, thereby reducing the machine learning algorithm effort in building a classifier. It also helps in improving the accuracy of the selected classifier for the dataset. There are different methods available for feature extraction such as a bag of words, word embeddings, TF-IDF and count vectors. However, in this study TF-IDF has been used [38]. StringToWordVector filter, which is available in WEKA, was applied.

Along with feature extraction, some other task of pre-processing has also been carried through the StringToWordVector filter, which is listed below:

- TF-IDF stands for Term Frequency-Inverse document frequency. It helps in identifying how important a certain word is to the document or a sentence. If a word has a high occurrence in a document, then its importance also increases, while this importance is offset by its frequency in a corpus [39]. For example, the word “کی”, “ہے”, “ہیں” etc. may appear very frequently in a document as well as in a complete corpus, while the word “خوبصورت” may occur frequently in a single document, but not in the whole corpus; thus, it would be considered relevant and important information for the analysis.
- LowerCaseToken parameter was used during the pre-processing of Roman Urdu only. All the words were transformed into lower case alphabets so that all the words were in the same format for further analysis.
- StopwordsHandler. In this parameter a file was given input which consisted of compiled stop words for Urdu (Nastaleeq) and Roman Urdu. The stop words were used in the respective pre-processing of Urdu and Roman Urdu.

- Tokenizer splits the words present in a string into a single token. These tokens are identified by WEKA using different delimiters. The delimiters used by the tool are `\r\n\t,,:\"'()?!"`
- WordsToKeep. This parameter allows the user to set the possible limit of words per class. It helps in keeping a maximum number of words even with the smaller frequency, which may be useful when building a classifier for the given dataset. In this study, the range was set as 10 000.

Other available parameters for StringToWordVector were not utilized in this study; hence, they were left with the default settings. The dataset was already pre-processed as per the requirement with the above-mentioned parameters. The last attribute of the loaded dataset in WEKA is considered the Class. However, in this case, after the StringToWordVector filter was applied to the dataset of Urdu and Roman Urdu tweets, the class was found to be the very first token. Similarly, the dataset was edited in WEKA and the column of the class was declared as "Attribute as Class" and was moved to the last attribute.

In the next step, another filter "NumericToBinary" was applied. This filter is used to normalize the attributes by converting each attribute into binary form, thereby assigning the value either 0 or 1. In the next step, another filter "Attribute Selection" was applied. The reason for applying this filter was to find the importance of attributes. Only applicable attributes are involved in the data mining process, instead of processing all the attributes. This way performance of the mining task is increased, and it will reduce the processing time as well. Thus, attribute selection algorithms are applied before implementing other data mining tasks. Attribute selection can be performed in two steps. The first step is subset generation, and the second step is ranking [40]. In this case, *InfoGainAttributeEval* was used followed by *Ranker* search. The training and testing dataset is passed through the previously mentioned pre-processing steps. Table IV represents the number of attributes obtained after the successful execution of pre-processing phase on the testing and training dataset.

The next step is to train these datasets with different machine learning algorithms in WEKA and then test the trained models.

TABLE I
EXAMPLE OF POLARITY ASSIGNMENT TO EXTRACTED URDU TWEETS

Urdu Tweet	English Translated Tweet	Pre-Processed English Tweet	Obtained Polarity	Polarity Label
دنیا کے مختلف حصوں میں کرونا کی صورتحال بہتر ہونے لگی	Corona's situation began to improve in different parts of the world	Corona's situation began to improve in different parts world	0.6	Positive
کرونا نے دنیا کی ساڑھے تین سو سال کی ترقی کو لپیٹ کر رکھ دیا	Corona wrapped up the world's three-and-a-half-year development	Corona wrapped world three half-year development	-0.166666667	Negative
یا اللہ اس کرونا کی وبا کو پاکستان سے ختم کر دے آمین	Oh Allah eradicate this epidemic from Pakistan. Amen	Allah eradicate epidemic Pakistan Amen	0	Neutral

TABLE II
EXAMPLE OF POLARITY ASSIGNMENT TO EXTRACTED ROMAN URDU TWEETS

Roman Urdu Tweet	English Translated Tweet	Pre-Processed English Tweet	Obtained Polarity	Polarity Label
ARY corruption ke khelaf awaz otane ka shukeria	ARY, thank you for speaking out against corruption	ARY thank you for speaking against corruption	0.7	Positive
Lanat ho aise jhoti media p	Damn such false media	Damn such false media	-0.4	Negative
Corrupt logon sy sawal tu hoga	Corrupt people will be questioned	Corrupt people questioned	0	Neutral

TABLE III
TESTING AND TRAINING DATASET OF URDU (NASTALEEQ) AND ROMAN URDU

	Positive Tweet	Negative Tweet	Neutral Tweet	Total	Training Dataset	Testing Dataset
Urdu (Nastaleeq)	586	379	949	1914	1414	500
Roman Urdu	378	300	449	1127	789	388

TABLE IV
NUMBER OF ATTRIBUTES OBTAINED AFTER PRE-PROCESSING IN WEKA TESTING AND TRAINING DATASETS

	Training Dataset		Testing Dataset	
	Number of Instances	Number of Attributes	Number of Instances	Number of Attributes
Urdu (Nastaleeq)	1414	3184	500	2158
Roman Urdu	789	3872	338	2193

After the training and testing datasets were prepared, the next step was the selection of a machine learning algorithm to be used for the sentiment analysis. Currently, the following are the two approaches that exist for machine learning:

- Supervised Machine Learning: Supervised learning can train the machine with the help of a well-labelled training dataset. The training dataset already has the instances tagged with the correct answers. Once the model is built, a testing set is supplied with unlabelled

instances. The trained model then produces the correct outcome for the testing dataset. Supervised learning can further be divided into classification in which an output label is a class such as rainy day, sunny day, patient, non-patient etc and other is regression in which the output label is a real-time numeric value such as price, weight, etc. [41].

- Unsupervised Machine Learning is based on the logic of building the model with training data that does not have any labelled class. It works by making groups of similar data, analysing the patterns, making some rules, etc. Unsupervised learning is further classified into clustering and association [41].

Thus, in this study supervised machine learning approach was adopted. Similarly, for this purpose classification algorithms were needed. WEKA provides a number of the classification algorithms. All of these algorithms were first trained by using the pre-processed training dataset. After that, the trained models were re-evaluated by supplying the test dataset.

Evaluation of a machine learning model is as important as building it on the desired dataset. Evaluation metrics identify the algorithm or model that is most suitable and robust for providing the solution to a given problem. Similarly, in this study, four parameters have been selected to find near to the best algorithm for sentiment analysis of Urdu (Nastaleeq) and Roman Urdu.

- Accuracy defines how effective a model is on the provided dataset. The ratio of correct predictions made to the total number of predictions is termed accuracy. The higher the accuracy of the classifier, the better the model is [42].

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total Number of predictions made}}.$$

In other words, it can also be said that the ratio of true positive and true negative to the total number of assessments is called accuracy. It is the proportion of the true results in a sample.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

where

TP = True Positive (Actual class is positive and so is the predicted class);

TN = True Negative (Actual class is negative and so is the predicted class);

FP= False Positive (Actual class is negative, but predicted class is positive);

FN = False Negative (Actual class is positive, but predicted class is negative).

- Precision is measured by dividing the true positive class in a model that was correct in the actual result also to the total number of predicted true positive and false positive. Precision has a range between 0 and 1, where 0 indicates low precision and 1 indicates high precision [43].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

- The recall is defined as the ratio of correctly identified positive values to the total number of actual values. Like precision, it also has a range between 0 and 1, where 1 is the highest range and 0 is the lowest. The more recall a value gets closer to the range of 1 the more a model is considered to yield the best results [43].

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

- F-Measure is a combined measure for precision and recall as it maintains the balance between both. It also takes into account both FN and FP as they should have minimum cost in the model to gain high accuracy [43].

$$F - \text{Measure} = \frac{2(\text{Precision} - \text{Recall})}{(\text{Precision} + \text{Recall})}.$$

IV. RESULTS

This section of the research gives a detailed view of the experiments carried out for the sentiment analysis of the Urdu language (Nastaleeq) and Roman Urdu. The results obtained from different classifiers on the dataset extracted from Twitter have been discussed.

A. Urdu (Nastaleeq)

All the existing classifiers in WEKA mentioned previously also were applied to the training dataset of Urdu (Nastaleeq) tweets to perform the sentiment analysis. The total number of the classifier is 31 and each of them is applied to the dataset. Execution of each classifier on the extracted dataset from Twitter played a significant role in identifying the most appropriate machine algorithm for the sentiment analysis, specifically in the Urdu language. During the experiment, it was observed that the results were obtained for 17 classifiers out of 31, while 14 classifiers were unable to give any results using WEKA.

The total number of instances for the training dataset was 1414, while for the testing it was 500. Once the classification model was built on the training data, the test data were provided to further predict the sentiment of the Urdu tweet. The most suitable algorithm that was found for this experiment was SMO (Sequential Minimal Optimization). This algorithm was anticipated in 1998 by John Platt. SMO is used for the training purpose of Support Vector Machine (SVM). The core purpose of SVM is to create boundaries between the classes involved in training and testing datasets, thereby considering the error risk involved in it. To achieve this goal, SVM requires the solution for large quadratic programming (QP) problems, which in turn requires plenty amount of time [44]. Similarly, SMO provides support to SVM by dividing the large QP problems into series of smaller QP chunks and then solving them analytically. This ability of SMO also reduces the time needed for training the dataset for SVM. Also, SMO can handle very large datasets easily as the amount of memory it requires is linear in the provided training data [45]. SMO had the highest accuracy of

83.6 % among other classifiers. 418 instances out of 500 were correctly classified by SMO, while only 82 instances were classified incorrectly. Based on the classification identified by SMO, it also satisfied the passing criteria of the evaluation

metrics as it had the precision value of 0.837; recall was obtained as 0.836 and *F*-measure as 0.833. The results obtained by all other classifiers are shown in Table V.

TABLE V
RESULTS OF MACHINE LEARNING ALGORITHM ON A DATASET OF URDU (NASTALEEQ) TWEETS FOR SENTIMENT ANALYSIS

Classifiers in WEKA	Total Test Instance	Correctly Classified Instances	Incorrectly Classified Instances	Total Accuracy %	Precision	Recall	F-Measure
Random Forest	500	413	87	82.6	0.849	0.826	0.820
Random Tree	500	386	114	77.2	0.770	0.772	0.769
REP Tree	500	283	217	56.6	0.555	0.566	0.521
LMT	500	N/A	N/A	N/A	N/A	N/A	N/A
J48	500	359	141	71.8	0.725	0.718	0.707
Hoeffding Tree	500	389	111	77.8	0.779	0.778	0.774
Decision Stump	500	N/A	N/A	N/A	N/A	N/A	N/A
BayesNet	500	392	108	78.4	0.798	0.784	0.775
NaiveBayes	500	N/A	N/A	N/A	N/A	N/A	N/A
NaiveBayesMultinomialText	500	N/A	N/A	N/A	N/A	N/A	N/A
NaiveBayesUpdateable	500	N/A	N/A	N/A	N/A	N/A	N/A
IBK	500	393	107	78.6	0.793	0.786	0.780
K-Star	500	N/A	N/A	N/A	N/A	N/A	N/A
LWL	500	N/A	N/A	N/A	N/A	N/A	N/A
Logistic	500	N/A	N/A	N/A	N/A	N/A	N/A
MultilayerPerceptron	500	N/A	N/A	N/A	N/A	N/A	N/A
SimpleLogisticRegression	500	408	92	81.6	0.819	0.816	0.814
SMO	500	418	82	83.6	0.837	0.836	0.833
AdaBoostM1	500	263	237	52.6	?	0.526	?
Bagging	500	341	159	68.2	0.694	0.682	0.664
AttributeSelectedClassifier	500	N/A	N/A	N/A	N/A	N/A	N/A
ClassificationViaRegression	500	259	241	51.8	?	0.518	?
CVParameterSelection	500	259	241	51.8	?	0.518	?
FiletredClassifier	500	359	141	71.8	0.725	0.718	0.707
IterativeClassifierOptimizer	500	285	215	57	0.562	0.570	0.527
LogitBoost	500	285	215	57	0.562	0.570	0.527
MultiScheme	500	259	241	51.8	?	0.518	41
MultiClassClassifierUpdateable	500	N/A	N/A	N/A	N/A	N/A	N/A
RandomCommitte	500	N/A	N/A	N/A	N/A	N/A	N/A
RandomizableFilteredClassifier	500	N/A	N/A	N/A	N/A	N/A	N/A
RandomSubSpace	500	N/A	N/A	N/A	N/A	N/A	N/A

It is evident from the graphical representation of the accuracy in Fig. 2 that the SMO algorithm had the highest accuracy compared to other classifiers, which were used in this study for the Urdu tweet dataset.

Also, it was able to classify a maximum number of instances correctly out of 500, while three algorithms, i.e., IBK, Logistic,

and Bagging, classified the least number of instances, which was 259. These three algorithms had the highest number of incorrectly classified instances (241).

SMO also satisfied the evaluation parameters. The results obtained for precision, recall, and *F*-Measure were near to the best for the supplied dataset.

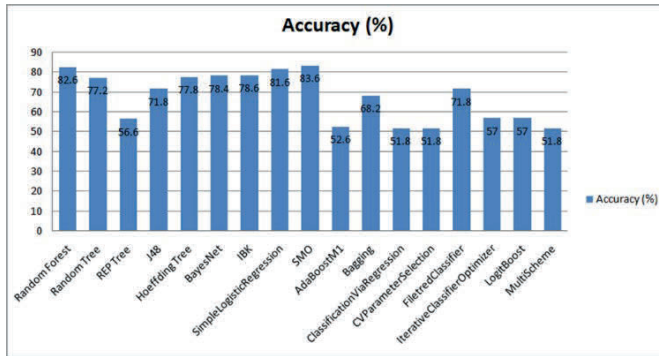


Fig. 2. Accuracy obtained for each classifier for Urdu tweets.

B. Roman Urdu

Considering the results of the sentiment analysis for Roman Urdu, training models were built with 31 different classifiers available in WEKA. The training dataset comprised 789 tweets with labelled sentiment class, i.e., Positive, Negative and Neutral, while the testing dataset had 388 tweets. However, out of these 31 classifiers, training models were successfully executed for 27 classifiers only. The remaining five classification algorithms were K-Star, LWL, MultilayerPerceptron, and AttributeSelectedClassifier. These algorithms in WEKA took a much longer time such as 60 minutes, but they failed to successfully execute on the given

dataset and hence no training model was obtained. During the experiment, it was also observed that precision and F-measure were not calculated by WEKA for the classifiers, which had an accuracy of less than 40 %. Such classifiers are DecisionStump, NaiveBayesMultinomialText, AdaBoostM1, ClassificationViaRegression, CVPParameterSelection and MultiScheme. Among all the applied classifiers, Random Forest was found to be the best for Roman Urdu. Random forest (RF) was proposed by Leo Breiman and Adèle Cutler in 2006. It is based on the technique of ensemble learning. Before random forest, there were only two classifiers available that worked on the logic of ensemble learning, i.e., boosting and bagging [46]. During the training phase, RF builds several decision trees also known as the forest of decision trees. Thus, the greater the forest, the highest the accuracy gained by the classifier. When it comes to predicting the data in the test dataset it works on the mechanism of voting, i.e., each of the created decision trees gives the class prediction vote for the test instance. Random forest assigns the class or category that has most of the votes [47]. It has achieved an accuracy of 89.0533 %. Also, it was able to classify 301 instances correctly out of 388 in the test dataset. The precision was 0.895, the recall was 0.891 and F-measure was 0.891. Table VI is the tabular representation of the results obtained for Roman Urdu sentiment analysis by WEKA.

TABLE VI
RESULTS OF MACHINE LEARNING ALGORITHM ON A DATASET OF URDU (NASTALEEQ) TWEETS FOR SENTIMENT ANALYSIS

Classifiers In WEKA	Total Test Instance	Correctly Classified Instances	Incorrectly Classified Instances	Total Accuracy %	Precision	Recall	F-Measure
Random Forest	388	301	37	89.0533	0.895	0.891	0.891
Random Tree	388	290	48	85.7988	0.860	0.858	0.859
REP Tree	388	176	162	52.071	0.531	0.521	0.515
LMT	388	292	46	86.3905	0.864	0.864	0.864
J48	388	229	109	67.7515	0.692	0.678	0.675
Hoeffding Tree	388	261	77	77.2189	0.808	0.772	0.771
Decision Stump	388	126	212	37.2781	?	0.373	?
BayesNet	388	269	69	79.5858	0.823	0.796	0.796
NaiveBayes	388	261	77	77.2189	0.808	0.772	0.771
NaiveBayesMultinomialText	388	118	220	34.9112	?	0.349	?
NaiveBayesUpdateable	388	261	77	77.2189	0.808	0.772	0.771
IBK	388	295	43	87.2781	0.882	0.873	0.874
K-Star	388	N/A	N/A	N/A	N/A	N/A	N/A
LWL	388	N/A	N/A	N/A	N/A	N/A	N/A
Logistic	388	290	48	85.7988	0.861	0.858	0.858
MultilayerPerceptron	388	N/A	N/A	N/A	N/A	N/A	N/A
SimpleLogisticRegression	388	273	65	80.7692	0.811	0.808	0.808
SMO	388	297	41	87.8698	0.879	0.879	0.879
AdaBoostM1	388	126	212	37.2781	?	0.373	?
Bagging	388	238	100	70.4142	0.715	0.704	0.702
AttributeSelectedClassifier	388	N/A	N/A	N/A	N/A	N/A	N/A
ClassificationViaRegression	388	118	220	34.9112	?	0.349	?
CVPParameterSelection	388	118	220	34.9112	?	0.349	?

Classifiers In WEKA	Total Test Instance	Correctly Classified Instances	Incorrectly Classified Instances	Total Accuracy %	Precision	Recall	F-Measure
FiletredClassifier	388	229	109	67.7515	0.692	0.678	0.675
IterativeClassifierOptimizer	388	138	200	40.8284	0.536	0.408	0.321
LogitBoost	388	171	167	50.5917	0.542	0.506	0.487
MultiScheme	388	118	220	34.9112	?	0.349	?
MultiClassClassifierUpdateable	388	298	40	88.1657	0.885	0.882	0.823
RandomCommitte	388	290	48	85.7988	0.861	0.858	0.858
RandomizableFilteredClassifier	388	288	50	85.2701	0.854	0.852	0.852
RandomSubSpace	388	215	123	63.6095	0.695	0.636	0.626

Two algorithms, i.e., Random Forest and MultiClassClassifierUpdateable were able to gain the highest accuracy, which was very close to each other, but Random Forest ranked first among all classifiers as it had an accuracy of 89.0533, a bit higher than MultiClassClassifierUpdateable, which had the accuracy of 88.1657. As it can be seen in Fig. 3, four algorithms out of 31 produced the worst results in terms of accuracy, which was 34.9112. These classifiers are NaïveBayesMultinomialText, ClassificationViaRegression, CVParameterSelection, MultiScheme.

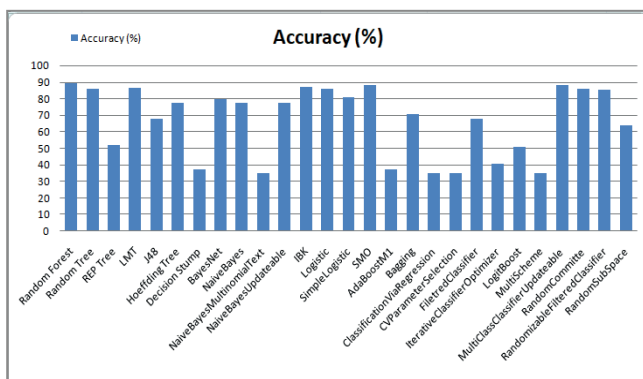


Fig. 3. Accuracy obtained for each classifier for Roman Urdu tweets.

Random forest yields the best results by classifying the highest number of tweets correctly in the testing dataset compared to other classifies.

V. DISCUSSION

Sentiment analysis has been performed in many languages to extract valuable information about products, brands, politics, etc. Such languages are very rich in sentiment analysis resources; the most important resource, in this case, is the availability of the lexicons, while other resources include stop words, techniques for tokenization, and stemming. Thus, unlike other languages, the Urdu language has poor resources for performing the sentiment analysis. The Urdu language lacks the availability of lexicons, similarly, the stemming of Urdu words gets more complicated. The most used python packages for the sentiment analysis (TextBlob and NLTK) work for 21 different languages, but they do not provide support for the Urdu language. This is because Urdu (Nastaleeq script) is very rich

in morphology; also, its grammar differs from English and other languages. Also, stop word removal is a very important task in the approach of sentiment analysis, i.e., lexicon-based or machine learning. However, there is no authentic resource available for the compiled stop words.

VI. CONCLUSION

This study aimed to perform the sentiment analysis of the national language of Pakistan (Urdu). Urdu is found on all platforms of social media and a wide range of its users use their national language (Urdu Script and Roman Urdu) to express their thoughts on different topics/issues going around the globe. Therefore, in this study, Twitter opted as the medium for data extraction in the form of Urdu (Nastaleeq and Roman Urdu) tweets. The data were extracted in different time intervals using Tweepy API. Also, Urdu is a resource-deprived language when it comes to the sentiment analysis compared to other rich languages, on which a wide variety of research has already been done for sentiment extraction using the lexicon-based approach. Especially, Roman Urdu has many irregularities in its spelling and allows for free-style writing. Thus, the Urdu language (Nastaleeq and Roman Urdu) lacks the lexicons and its associated components such as stemming, tokenization, and lemmatization. Another step in the research of sentiment analysis specifically for the Urdu language can be the creation of lexicons, which might yield better results than other available approaches. Due to this reason, a machine learning approach was adopted to perform the sentiment analysis in this study.

Once the data were successfully extracted and pre-processed, each tweet was assigned the polarity and 31 different classifiers were applied to the obtained dataset. The best one was chosen for Urdu and Roman Urdu based on the evaluation metrics. SMO was found best for the sentiment analysis on Urdu (Nastaleeq) and Random Forest ranked number one for Roman Urdu. Also, the stop word list has been compiled for Urdu script from three different sources, while stop words were extracted manually from the dataset for Roman Urdu. The training and then testing of the classifiers took place in the tool WEKA. These lists of stop words can play a significant role in future research on the Urdu language. Moreover, in this study, some of the classifiers out of 31 were unable to give any results in the WEKA tool on the provided dataset. Thus, in the future, a different approach and techniques of preprocessing can be

adopted on the dataset so that some valuable results can be obtained from the leftover machine learning algorithms as well.

REFERENCES

- [1] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma, "Sentiment analysis: A review and comparative analysis of web," *Information Sciences*, vol. 311, pp. 18–38, Aug. 2015. <https://doi.org/10.1016/j.ins.2015.03.040>
- [2] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs data mining and knowledge discovery*, vol. 8, no. 4, July 2018. <https://doi.org/10.1002/widm.1253>
- [3] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzivasvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, pp. 214–224, Mar. 2017. <https://doi.org/10.1016/j.eswa.2016.10.043>
- [4] K. K. Mohbey, B. Bakariya, and V. Kalal, "A study and comparison of sentiment analysis techniques using demonetization: Case study," in *Sentiment Analysis and Knowledge Discovery in Contemporary Business*, 2018, pp. 1–14. <https://doi.org/10.4018/978-1-5225-4999-4.ch001>
- [5] C. S. Khoo and S. B. Johnkhan, "Lexicon-based sentiment analysis: Comparative Evaluation of Six Sentiment Lexicons," *Journal of Information Science*, vol. 44, no. 4, pp. 491–511, 19 Apr. 2017. <https://doi.org/10.1177/0165551517703514>
- [6] N. Boudad, R. Faizi, R. O. Haj Thami, and R. Chiheb, "Sentiment analysis in Arabic: A review of the literature," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2479–2490, Dec. 2018. <https://doi.org/10.1016/j.asej.2017.04.007>
- [7] S. Tartir and I. A. Nabi, "Semantic sentiment analysis in Arabic social media," *Journal of King Saud University – Computer and Information Sciences*, vol. 29, no. 2, pp. 229–223, Apr. 2017. <https://doi.org/10.1016/j.jksuci.2016.11.011>
- [8] A. K. Rathore, V. Ilavarasan, and Y. K. Dwivedi, "Social media content and product co-creation: An emerging paradigm," *Journal of Enterprise Information Management*, vol. 29, no. 1, pp. 7–18, Feb. 2016. <https://doi.org/10.1108/JEIM-06-2015-0047>
- [9] J. L. Sheela, "A review of sentiment analysis in Twitter data using Hadoop," *International Journal of Database Theory and Application*, vol. 9, no. 1, pp. 77–86, 2016. <https://doi.org/10.14257/ijda.2016.9.1.07>
- [10] S. A. Salloum, M. Al-Emran, A. A. Monem, and K. Shaalan, "A survey of text mining in social media: Facebook and Twitter perspectives," *Advances in Science, Technology and Engineering Systems*, vol. 2, no. 1, pp. 127–133, 2017. <https://doi.org/10.25046/aj020115>
- [11] "Twitter launches," A&E Television Networks, 14 July 2020. [Online]. Available: <https://www.history.com/this-day-in-history/twitter-launches>. Accessed on: Aug. 2020.
- [12] "Number of monetizable daily active Twitter users (mDAU) worldwide from 1st quarter 2017 to 2nd quarter 2020," 23 July 2020. [Online]. Available: <https://www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/>. Accessed on: Aug. 2020.
- [13] Y. Lin, "10 Twitter statistics every marketer should know in 2022 [infographic]," 30 July 2019. [Online]. Available: <https://www.oberlo.com/blog/twitter-statistics>. Accessed on: Oct. 2019.
- [14] D. Hattem and L. Lomicka, "What the Tweets say: A critical analysis of Twitter research in language learning from 2009 to 2016," *E-Learning and Digital Media*, vol. 13, pp. 5–23, Oct. 2019. <https://doi.org/10.1177/2042753016672350>
- [15] Twitter Inc., "Twitter for websites-supported languages," 2019. [Online]. Available: <https://developer.twitter.com/en/docs/twitter-for-websites/twitter-for-websites-supported-languages/overview>. Accessed on: 2019.
- [16] H. B. Zaya, A. A. Raza, and A. Ather, "Urdu word segmentation using conditional random fields (CRFs)," in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico: Association for Computational Linguistics, 2018, pp. 2562–2569.
- [17] V. S. Pagolu, K. N. R. Challa, and G. Panda, "Sentiment analysis of Twitter data for predicting stock market movements," in *International conference on Signal Processing, Communication, Power and Embedded System*, Paralakhemundi, India, Oct. 2016, pp. 1345–1350. <https://doi.org/10.1109/SCOPES.2016.7955659>
- [18] R. P. Schumaker, A. T. Jarmosko, and J. L. S. Chester, "Predicting wins and spread in the Premier League using a sentiment analysis of twitter," *Decision Support Systems*, vol. 88, pp. 76–84, Aug. 2016. <https://doi.org/10.1016/j.dss.2016.05.010>
- [19] D. Pope and J. Griffith, "An analysis of online Twitter sentiment surrounding the European," in *8th International Conference on Knowledge Discovery and Information Retrieval*, Porto, Portugal, 2016, pp. 299–306. <https://doi.org/10.5220/0006051902990306>
- [20] A. C. Pandey, D. S. Rajpoot, and M. Saraswat, "Twitter sentiment analysis using hybrid cuckoo search method," *Information Processing & Management*, vol. 53, no. 4, pp. 764–779, July 2017. <https://doi.org/10.1016/j.ipm.2017.02.004>
- [21] H. K. Aldayel and A. M. Azmi, "Arabic tweets sentiment analysis – a hybrid scheme," *Journal of Information Science*, vol. 42, no. 6, pp. 782–797, Oct. 2016. <https://doi.org/10.1177/016555151610513>
- [22] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Arabic language sentiment analysis on health services," in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, Nancy, France, Apr. 2017, pp. 114–118. <https://doi.org/10.1109/ASAR.2017.8067771>
- [23] M. Heikal, M. Torki, and N. El-Makky, "Sentiment analysis of Arabic Tweets using deep learning," *Procedia Computer Science*, vol. 142, pp. 114–122, 2018. <https://doi.org/10.1016/j.procs.2018.10.466>
- [24] A. Hassan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for Twitter accounts," *Mathematical and Computational Applications*, vol. 23, no. 1, Feb. 2018. <https://doi.org/10.3390/mca23010011>
- [25] I. Javed, H. Afzal, A. Majeed, and B. Khan, "Towards creation of linguistic resources for bilingual sentiment analysis of Twitter data," in *International Conference on Applications of Natural Language to Data Bases/Information Systems*, Jun. 2018. https://doi.org/10.1007/978-3-319-07983-7_32
- [26] S. Ahmed, S. Hina, and R. Asif, "Detection of sentiment polarity of unstructured multi-language text from social media," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 7, pp. 199–203, 2019. <https://doi.org/10.14569/IJACSA.2018.090728>
- [27] T. R. Soomro and S. M. Ghulam, "Current status of urdu on Twitter," *Sukkur IBA Journal of Computing and Mathematical Sciences*, vol. 3, no. 1, pp. 28–33, 2019. <https://doi.org/10.30537/sjcms.v3i1.397>
- [28] F. Noor, M. Bakhtyar, and J. Baber, "Sentiment analysis in E-commerce using SVM on Roman Urdu text," in *International Conference for Emerging Technologies in Computing*, Jul. 2019. https://doi.org/10.1007/978-3-030-23943-5_16
- [29] H. Ghulam, F. Zeng, W. Li, and Y. Xiao, "Deep learning-based sentiment analysis for Roman Urdu text," in *2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018*, vol. 147, 2018, pp. 131–135. <https://www.sciencedirect.com/journal/procedia-computer-science/vol/147/suppl/C>
- [30] Z. Mehmood et al., "Deep sentiments in Roman Urdu text using recurrent convolutional neural network model," *Information Processing and Management*, vol. 57, no. 4, Feb. 2020, Art no. 102233. <https://doi.org/10.1016/j.ipm.2020.102233>
- [31] V. Bonta, N. Kumares, and J. N., "A comprehensive study on lexicon based approaches for sentiment analysis," *Asian Journal of Computer Science and Technology*, vol. 8, no. S2, pp. 1–6, Mar. 2019. <https://doi.org/10.51983/ajcst-2019.8.S2.2037>
- [32] S. Sarica and J. Luo, "Stopwords in technical language processing," *PLoS ONE*, vol. 16, no. 8, Aug. 2021, Art no. e0254937. <https://doi.org/10.1371/journal.pone.0254937>
- [33] K. S. Dar, A. B. Shafat, and H. U. Muhammad, "An efficient stop word elimination algorithm for Urdu language," in *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, Thailand, Jun. 2017. <https://doi.org/10.1109/ECTICon.2017.8096386>
- [34] M. Usman, S. Ayub, Z. Shafique, and K. Malik, "Urdu text classification using majority voting," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 8, pp. 265–273, 2016. <https://doi.org/10.14569/IJACSA.2016.070836>
- [35] K. Riaz and D. Becker, "Stopword identification in an Urdu corpus".
- [36] A. Burney, B. Sami, N. Mahmood, Z. Abbas, and K. Rizwan, "Urdu text summarizer using sentence weight algorithm for word processors," *International Journal of Computer Applications*, vol. 46, no. 19, pp. 38–43, May 2012.

- [37] E. D. P. Kaur and E. P. Singh, "A comparative research of rule based classification on dataset using WEKA TOOL," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 9, Sep. 2019. chrome-extension://efaidnbmnnpbpcjpcglefindmkaj/https://www.irjet.net/archives/V6/i9/IRJET-V6I9345.pdf
- [38] R. Ahuja, A. Chuga, S. Kohlia, S. Gupta, and P. Ahuja, "The impact of features extraction on the sentiment analysis," in *International Conference on Pervasive Computing Advances and Applications*, vol. 152, 2019, pp. 341–348. <https://www.sciencedirect.com/journal/procedia-computer-science/vol/152/suppl/C>
- [39] B. Stecanella, "What is TF-IDF?" May 2019. [Online]. Available: <https://monkeylearn.com/blog/what-is-tf-idf/>. Accessed on: July 2020.
- [40] S. Gnanambal, M. Thangaraj, V. T. Meenatchi, and V. Gayathri, "Classification algorithms with attribute selection: an evaluation study using WEKA," *International Journal of Advanced Networking and Applications*, vol. 9, no. 6, pp. 3640–3644, May 2018.
- [41] M. Desai and M. A. Mehta, "Techniques for sentiment analysis of Twitter data: A comprehensive survey," in *International Conference on Computing, Communication and Automation*, Greater Noida, India, Apr. 2016, pp. 149–154. <https://doi.org/10.1109/CCAA.2016.7813707>
- [42] S. Yıldırım, "How to best evaluate a classification model," 17 March 2020. [Online]. Available: <https://towardsdatascience.com/how-to-best-evaluate-a-classification-model-2ed12bcc587>.
- [43] P. Subedi, "Machine learning – The different ways to evaluate your classification models and choose the best one!" 18 August 2020. [Online]. Available: <https://medium.com/kharpann/machine-learning-the-different-ways-to-evaluate-your-classification-models-and-choose-the-best-1281542432c>. Accessed on: July 2020.
- [44] M. Ghosh and G. Sanyal, "An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning," *Journal of Big Data*, vol. 5, Nov. 2018, Art no. 44. <https://doi.org/10.1186/s40537-018-0152-5>
- [45] V. Chaurasia and S. Pal, "A novel approach for breast cancer detection using data mining techniques," *International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization)*, vol. 2, no. 1, pp. 2456–2465, Jul. 2017. https://www.researchgate.net/publication/259979477_A_Novel_Approach_for_Breast_Cancer_Detection_using_Data_Mining_Techniques
- [46] Y. A. Amrani, M. Lazaar, and K. E. E. Kadir, "Random forest and support vector machine based hybrid approach to sentiment analysis," in *The First International Conference on Intelligent Computing in Data Sciences*, vol. 127, 2018, pp. 511–520. <https://www.sciencedirect.com/journal/procedia-computer-science/vol/127/suppl/C>
- [47] M. A. Fauzi, "Random forest approach for sentiment analysis in Indonesian language," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 1, pp. 46–50, Oct. 2018. <https://doi.org/10.11591/ijeecs.v12.i1.pp46-50>

Iffraah Rehman received B.Sc. and M.Sc. degrees in Computer Science from the Institute of Business Management (IoBM), Karachi, Sindh, Pakistan. She is currently serving as a Software Developer, ISD at State Bank of Pakistan. She is an experienced Research Database Officer with a demonstrated history of working in the facilities services industry. Skilled in PHP, JavaScript, MYSQL & SQL SERVER, Web Application, C# / ASP.NET, Oracle 11g Forms / Reports, SQL/PLSQL, WordPress and Microsoft Office. She is a strong information technology professional in Web and Database. E-mail: std_23371@iobm.edu.pk

Tariq Rahim Soomro (Senior Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in Computer Science from the University of Sindh, Jamshoro, Sindh, Pakistan, and the PhD degree in Computer Applications from Zhejiang University, Hangzhou, Zhejiang, China. He is currently a Professor of Computer Science, the Dean of the College of Computer Science and Information Systems, and an Acting Rector with the Institute of Business Management (IoBM), Karachi, Sindh, Pakistan. He has more than 28 years of extensive and diverse experience as an administrator, computer programmer, researcher, and teacher. As an administrator, he served as a coordinator, the head of the department, the head of the faculty, the dean of the faculty, the head of the academic affairs, an acting rector, and has wide experience in accreditation related matters, including ABET, USA, NCEAC, HEC Pakistan, and the Ministry of Higher Education and Scientific Research, United Arab Emirates (UAE).

He is also an IEEE Computer Society Distinguished Visitor (2021–2023). He is a member of the Task Force on Arabic Script IDNs by the Middle East Strategy Working Group (MESWG) of ICANN. He has been a member of the Project Management Institute (PMI) since 2007, and ACM since 2019. He has been a Senior Member of the IEEE Computer Society since 2005, and the International Association of the Computer Science and Information Technology (IACSIT) since 2012. He has been a Life Member of the Computer Society of Pakistan (CSP) since 1999, and a Global Member of the Internet Society (ISOC), USA since 2006. He has been an Active Member of the IEEE Karachi Section (Region 10). He is also serving as the member of the Executive Committee (ExCom) (2017–2022), the IEEE R10 Southern Area Coordinator Computer Society (2020–2022), and the IEEE R10 Education Activity Committee (EAC) (2021). He received the ISOC Fellowship to the IETF for the 68th Internet Engineering Task Force (IETF) Meeting. He served as the Secretary for the Karachi Section (2018–2019), the Chair for the GOLD Affinity Group (2002), a member for the Executive Committee, and a branch councillor (2002 & 2016). He is currently serving as the Vice-Chair Karachi Section (2020–2023).

E-mail: tariq.soomro@iobm.edu.pk

ORCID iD: <https://orcid.org/0000-0002-7119-0644>