Article

Machine Learning-Based Sentiment Analysis for Twitter Accounts

Ali Hasan [1], Sana Moin [1], Ahmad Karim [2] and Shahaboddin Shamshirband [3,4,*] iD

[1] Department of Computer Science, Air University, Multan Campus, Multan 60000, Pakistan; alihasan1520@gmail.com (A.H.); sanamoin7@gmail.com (S.M.)
[2] Department of Information Technology, Bahauddin Zakariya University, Multan 60000, Pakistan; ahmadkarim@bzu.edu.pk
[3] Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[4] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
* Correspondence: shahaboddin.shamshirband@tdt.edu.vn

Abstract: Growth in the area of opinion mining and sentiment analysis has been rapid and aims to explore the opinions or text present on different platforms of social media through machine-learning techniques with sentiment, subjectivity analysis or polarity calculations. Despite the use of various machine-learning techniques and tools for sentiment analysis during elections, there is a dire need for a state-of-the-art approach. To deal with these challenges, the contribution of this paper includes the adoption of a hybrid approach that involves a sentiment analyzer that includes machine learning. Moreover, this paper also provides a comparison of techniques of sentiment analysis in the analysis of political views by applying supervised machine-learning algorithms such as Naïve Bayes and support vector machines (SVM).

Keywords: Twitter; sentiment analyzer; machine learning; WordNet; word sequence disambiguation (WSD); Naïve Bayes

1. Introduction

In recent years, a huge number of people have been attracted to social-networking platforms like Facebook, Twitter and Instagram. Most use social sites to express their emotions, beliefs or opinions about things, places or personalities. Methods of sentiment analysis can be categorized predominantly [1] as machine-learning [2], Lexicon-based [3] and hybrid [4,5]. Similarly, another categorization has been presented [6] with the categories of statistical, knowledge-based and hybrid approaches. There is a space for performing challenging research in broad areas by computationally analyzing opinions and sentiments [7]. Therefore, a gradual practice has grown to extract the information from data available on social networks for the prediction of an election, to use for educational purposes, or for the fields of business, communication and marketing. The accuracy of sentiment analysis and predictions can be obtained by behavioral analysis based on social networks [8].

To reveal the views of the leaders of two big democratic parties in India [9], data was collected from the public accounts of Twitter. Opinion Lexicon [10] was used to find a total number of positive, neutral and negative tweets. Findings show that analyzing the public views could help political parties transform their strategies.

A keyword-based tweet collection, focused on the names of the political parties and political celebrities of Pakistan [11], was made to test the popularity of the party for the elections of 2013. This dataset was tested with both supervised and unsupervised machine-learning algorithms. It used the Rainbow tool [12] and applied Prind, K nearest neighbors (KNN) [13], and Naïve Bayes

(NB) [14] classification methods on unigram data. The same dataset was tested using supervised machine-learning algorithms which were support vector machines (SVM), NB, random forest (RF) [15] and Naïve Bayes multinomial NBMN [16]. For the purpose of smoothing the data by removing zero values, it used Laplace and Porter stemmer [17,18]. Term frequency–inverse document frequency (TF–IDF) [19] was applied for the purpose of finding strongly related words for relevant documents. It also applied 5-fold cross validation using the Waikato Environment for Knowledge Analysis (Weka) [20,21]. The main purpose of selecting Twitter's profile data is that we can get qualitative information from this platform because Twitter contains the authenticated accounts of politicians, which is not the case of Facebook or Instagram etc. Additionally, by contrast with Facebook, Twitter restricts users to give their compact and complete opinions in 280 characters.

Recent studies have proven [22,23] that with Twitter it is possible to get people's insight from their profiles in contrast to traditional ways of obtaining information about perceptions. Furthermore, authors of [24] proposed an algorithm for exploiting the emotions from tweets while considering a large scale of data for sentiment analysis. To identify social communities with influential impact, a novel method was proposed by [25] and implemented by assigning metric value to each of the user's emotional posts. Subsequently, the contribution of this paper includes the analysis of election sentiments gathered from Twitter profiles, with various sentiment analyzers. In addition, this paper presents the validation of results obtained from each analyzer with machine-learning classifiers. Our analysis is based on the comparison of different sentiment analyzers and validates the results with the different classifiers. The experiment on Twitter data will show which technique has a better capability of measuring sentiment prediction accuracy. The remainder of the paper is organized in the following sections. Section 2 covers the discussion of related searches in the domain of sentiment analysis of Twitter data including the accuracy of approaches adopted to predict election results. The proposed framework for the validation of sentiment analysis is discussed in Section 3. Section 4 covers the detail from the collection of data to the sentiment analysis, and debates the experiment we conducted on the data-set using sentiment analyzers and classifiers. Furthermore, the accuracies obtained are discussed. This paper concludes in Section 5.

2. Background Study/Literature Survey

The comparison of two politicians was made on the basis of real-time Twitter data [26], extracted from Twitter by using Twitter-streaming application programming interface (API) [27]. Two sentiment analyzers named SentiWordNet [28] and WordNet [29] were used to find positive and negative scores. To add accuracy to the model, negation handling [30] and word sequence disambiguation (WSD) were used [31]. Twitter streaming API was also used to gather data by the authors of [32] for the prediction of the Indonesian presidential elections. The aim was to use Twitter data to understand public opinion. For this purpose, after the collection of data, the study performed automatic buzzer detection to remove unnecessary tweets and then analyzed the tweets sentimentally by breaking each tweet into several sub-tweets. After that, it calculated sentiment polarity and, to predict election outcome, used positive tweets associated with each candidate, and then used mean absolute error (MAE) [33] in order to measure the performance of the prediction and make the claim that this Twitter-based prediction was 0.61% better than the same type of surveys conducted traditionally.

To forecast a Swedish election outcome, other than sentiment analysis a link structure was analyzed using Twitter [34] involving politicians' conversations. For this purpose, this used a link-prediction [35] algorithm and showed that account popularity known by structural links have more similarities with outcomes of the vote. This also revealed that there is a strong dependence between the standing of selected politicians and the outcome of the general election and also of the official party account and European election outcome. A methodology was created to test Brazilian municipal elections in 6 cities [22]. In this methodology, sentiment analysis was taken into consideration along with a stratified sample [36] of users in order to compare the characteristics of the findings with actual voters.

As the sentiment analysis of tweets has gained popularity in recent years, the sentiments of queries generated by users has been calculated [37] by page-rank algorithms and the Naïve Bayes classifier. For the prediction of the 2016 US elections, manually annotated corpus-based hashtags [38] along with negation detection were tested and a claim was made that a 7% accuracy level had increased. The model adopted to rank the candidates of political parties [38] included the lexicon-based approach [39] and the linguistic inquiry word count (LIWC) [40].

Although a lot of work has been done to analyze tweet sentiments by applying techniques of machine learning [2,41], the work of [42] was based on a set of techniques of machine learning that were Naïve Bayesian, SVM [43] and entropy-based [44] along with semantic analyses to classify product reviews or sentences. In [45] a rigorous dataset was constructed to determine and politically rank individuals for the US 2010 midterm elections, based on the political discussion and network-based data available on their Twitter timelines. Features to be analyzed were split into two broad categories: (a) user-level based on content; and (b) network-level, to determine relationships among the users. This used many methods including SVM for politically classifying individuals and to determine percentage accuracy of the methods adopted.

On the other hand, research has been conducted [46] based on the objective of shifting traditional research approaches to Twitter as a new source of information on an electoral campaign and tracking the behavior of people's perception as campaigns develop over time. To report the factors of personality, party and policy a huge amount of Twitter data was collected from 6 presidential campaigns for the U.S. presidential elections in 2016 but this did not acknowledge the authenticity of accounts.

For the estimation of political affiliations among users, the Twitter REST API [47] was used to crawl the list of followers for landmarks of the Republican and Democrat campaigns. This used SentiWordNet lexicon [28,48] to classify the tweets as negative or positive on the basis of comprehensive sentiment scores. A hybrid classification-based algorithm along with a Twitter opinion-mining framework was presented [49] to analyze and classify the Twitter feed with improved accuracy. The algorithm presented for polarity classification in the framework consisted of the enhanced emoticon classifier (EEC), improved polarity classifier (IPC) and SentiWordNet classifier (SWNC). The accuracy obtained from the polarity classification algorithm (PCA) [49] was 85.7% when compared with other similar approaches.

Additionally, a hybrid approach was adopted [8] that involved supervised classifiers which were the artificial neural networks, feed-forward support vector machines (SVM), maximum entropy and Naïve Bayes. To predict election results, they proposed an approach of user influence factor exploitation. Understanding influential factors and analyzing data with SVM resulted in approximately 88% accuracy in both the 2013 Karnataka assembly elections and the 2012 US presidential elections.

To show that Twitter trends play an important role in electoral sentiments, the authors of [50] collected hashtag-based tweets covering the candidates of the Indian elections 2014. They did not include neutral tweets for the analysis because they found these kinds of tweets problematic for sentiment analyses that are in favor of more than one party. Two lexicons [51,52] were combined for the sentiment analysis of tweets. This bipolar lexicon was best in the case of the analysis of two parties, but for the classification of multiple parties they created variables and their approach was not sufficiently state-of-the-art to calculate sentiment score when more parties were involved in the analysis.

Authors of [53] used Textblob for pre-processing, polarity, the polarity confidence calculation, and they validated the obtained results by SVM and Naïve Bayes using Weka; they reported the highest accuracy of Naïve Bayes with a 65.2% rate, which was 5.1% more than the SVM accuracy rate. An unsupervised machine-learning algorithm was introduced [54] to rate the reviews as thumbs up and thumbs down; 410 reviews were gathered from Epinions and tested, the observed accuracy of the algorithm was 74%. Moreover, for election prediction, authors of [55] tested sentiments related to political parties by SVM and Naïve Bayes. Consequently, they predicted the possibilities of the BJP winning the elections in 2016 on the basis of the SVM result with 78.4% accuracy, which was 16.3%

greater than the Naïve Bayesian results. Additionally, a methodology was proposed [56] for better forecasting of elections based on public opinion from web platforms, which resulted in 83% of accuracy.

In this paper, we are not aiming to propose any sentiment-prediction technique, but analyzing the notable sentiment techniques in election domains. For this purpose, the dataset is based on users' tweets about politics across Pakistan. These kinds of analyzers are also used in other domains like health, disease and personality prediction. To the best of our knowledge, this paper is first attempt to cater Urdu tweets and further translate them into English for sentiment analysis. In addition, our results from analyzers are hence validated from statistical machine-learning classifiers like Naïve Bayes and SVM in Weka. Furthermore, we have demonstrated the outcomes of both: sentiment analyzers and their machine-learning validation using a tabular format to provide a clear view about the accuracy percentages obtained.

3. Framework

We have presented our framework in which we have explained a process from the collection, sentiment analysis, and classification of Twitter opinions. We considered tweets that were posted by users in the form of hashtags to express their opinions about current political trends. We then stored the retrieved tweets in the database, translated Urdu tweets, and pre-processed the dataset. After pre-processing, the remaining data was split into the training set with 1690 tweets and the test set with 400 tweets. Polarity and subjectivity were calculated using three different libraries, SentiWordNet, W-WSD and TextBlob. We applied the Naïve Bayes and SVM classifier on training set in Weka and built a classification model. The model was tested on the training dataset to obtain the accuracy result of each classifier. The following sub sections detail the tools and techniques that aided us in sentiment analysis. Furthermore, a clear view of the sentiment-analysis framework is illustrated below in Figure 1a,b.
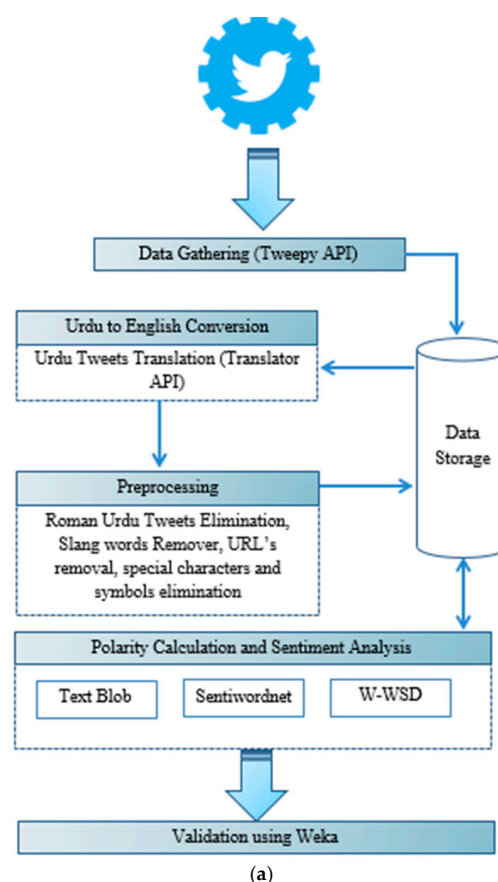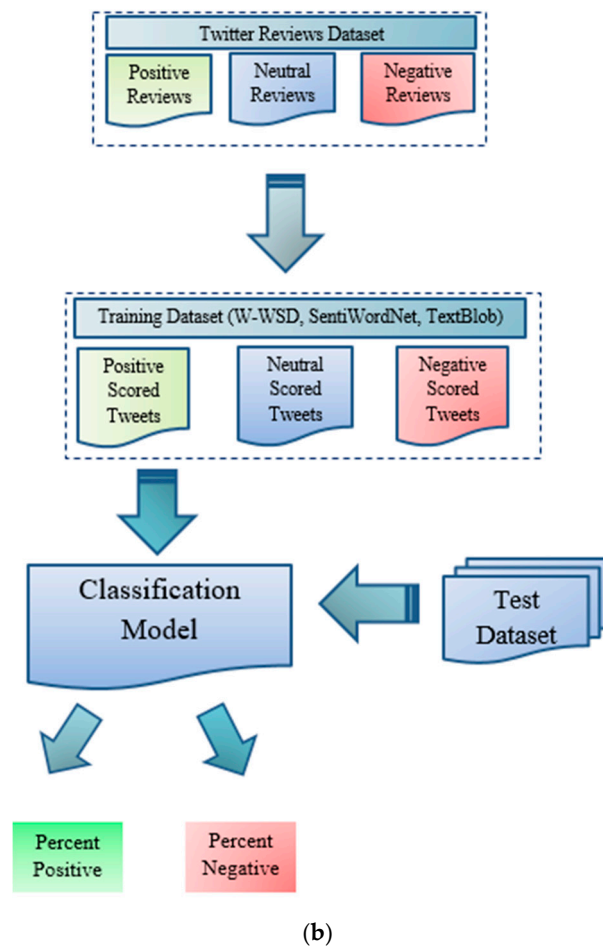

(a)

Figure 1. *Cont.*

(**b**)

Figure 1. (**a**) Framework of tweet classification and sentiment analysis; (**b**) framework of tweet classification and sentiment analysis.

A- Data Gathering

To gather public opinion based on collected hashtags related to views about political parties including Twitter top trends, we used Tweepy API [27,57]. We have created an account on Tweepy API linked to our Twitter account. To retrieve the tweets, Tweepy API accepts parameters and provides the Twitter account's data in return.

Retrieved tweets, from Twitter accounts, were saved in the database under the following fields: *twitter_id, hashtag, tweet_created, user_id, screen_name, tweet_text, retweet_count, follower_count, and favourite_count* of each tweet. The gathered tweets numbered 100,000.

We select hashtags that were trending on Twitter, representing the political views of people. In addition, we searched for some more hashtags of a political nature. Tweets were retrieved against the hashtags, some of which are represented below in Table 1.

Table 1. Hashtags.

#JITReport	#GameOverNawaz	#WellDoneJIT	تینوں_نسلیں_چور_نکلیں#
#Calibri	#JITFinalReport	#IfMaryamIsInnocent	#BahadurBetiMaryam
#CrimeMinisterNawaz	#JiyeBhutto	#MaryamMeriAwaz	#PakStandsWithSC
#IK	#PoorMasses	#Election2018	#MaryamQaumKiBetiHai
#SAYNOTOCORRUPTION	#SlavesOfMaryamNawaz	#KulbhushanJadhav	#RainExposesMayor

Function 1 below, presents the steps for gathering tweets using Tweepy API for hashtags.

Function 1: F1 (U, TA, HL, O)	
Input Attributes:	**Output Attribute:**
U: Registered Twitter developer profile for OAuthHandler consumer key	**O:** Set of tweets in response to input attributes for each hash tag and stored in database.
TA: Tweepy API was used to get tweets from the hash tag.	
HL: List of Hashtags	

```
function F1 (U, TA, HL)
      for all htag from HL do
             auth = TA.OAuthHandler(U);
             auth.set_access_token(access key, access_secret);
             api = tweepy.API(auth);
             api.wait_on_rate_limit = True;
             api.wait_on_rate_limit_notify = True;
             for all tweet from
             tweepy.Cursor(api.search,q=HashTag,geocod='30.3753,69.3451',
             since_id=Get_latest_tweet_ID(htag)).items() do
                   store_to_database (tweet.retweet_count,
                   tweet.favorite_count,tweet.id,HashTag,tweet.created_at, tweet.text,tweet.user.id,
                   tweet.user.followers_count,tweet.user.screen_name
             end for
      end for
end function
```

B- Urdu-to-English Conversion

Tweets gathered from user accounts were in the Urdu, English, and Roman Urdu Language. To add novelty in data analysis, we have included tweets of the Urdu language by taking into account the fact that Urdu is the national language of Pakistan. So, people prefer to present their views in their native language. Before obtaining sentiments, it is necessary to convert data into a format so that they can be accurately classified as negative, neutral or positive. For this purpose, we first translated Urdu tweets into English. To translate Urdu tweets into English, we ran Translator API. An example of the conversion is as follows.

"انصاف ہو کس طرح کی دل صاف نہیں ہے

دل صاف ہو کس طرح کی انصاف نہیں ہے"

"The heart is not clear how to be justice The heart is not clear how justice is".

C- Data Pre-processing

In the pre-processing step, we removed the irrelevant Twitter data. First, we just kept tweets that were in English and Urdu in the data storage. After the translation of Urdu tweets into English, the remaining tweets belonging to any other language were removed. To remove Roman Urdu tweets, we searched for strings including ther most common words used in Roman Urdu e.g., main, ki, han, ko, kaisy, hum, ap, hy, hai, thi, na, and removed all strings containing these types of words. An example of a Roman Urdu Tweets is "#JIT ko appreciate krna zrori hai. Na biky na jhuky na dary". Furthermore, we removed URLs from tweets, because URLs direct to extra information that was not a requirement for sentiment analysis in our approach. The mention of other accounts with the @ sign were also removed including any other symbol or special character such as " ' ? ! ; : # $ % & ( ) * + − / < > = [ ] \ ^ _ { } | ~.

Function 2 was created to add the capability of converting Urdu tweets into English, and to perform other tasks for pre-processing.

Function 2: F2 (T, CT, PT)	
Input Attributes:	**Output Attributes:**
T: Tweet from Database	**CT:** Converted tweets from Urdu language to English using language translator API [58].
	PT: Pre-processed tweets by eliminating unwanted words like symbols, special characters and other-than-English language words using StopWordRemoval and REGEXP. (Regular Expression)

```
function F2(T)
    for all tweet from T do
        language = check_language(tweet);
        if language = "ur" then
            tweet = translate_to_english(tweet);
        end if
        if  tweet.words exist in roman_urdu_regexpList{}
            delete_from_database(tweet);
        else
            preprocessing(tweet);
        end if
    end for
```

D- Polarity Calculation and Sentiment Analysis

Sentiment analysis can provide valuable insights from social media platforms by detecting emotions or opinions from a large volume of data present in unstructured format. Sentiment analysis includes three polarity classes, which are negative, neutral and positive. The polarity of each tweet is determined by assigning a score from -1 to 1 based on the words used, where a negative score means a negative sentiment and a positive score means a positive sentiment while the zero value is considered a neutral sentiment. A score of subjectivity assigned to each tweet is based on whether it is representing a subjective meaning or an objective meaning; the range of subjectivity score is also from 0 to 1 where a value near to 0 represents objective and near to 1 subjective.

For detecting the polarity and subjectivity of political reviews, and to give a clear view of the most accurate analyzer for the polarity and subjectivity calculator, we used Textblob, SentiWordNet and Word Sense Disambiguation (WSD) sentiment analyzers. Textblob [59] comes with the basic features of natural-language processing essentials; we used this analyzer for the polarity and subjectivity calculation of tweets. Similarly, we used SentiWordNet [60], which is a publicly available analyzer of the English language that contains opinions extracted from a wordnet database. In addition to that, W-WSD [31,61] has the ability to detect the correct word sense within a specified context. Baseline words/Unigrams are clear representatives for the calculation of polarity [56]. Consequently, we used Unigram data in W-WSD. Despite admitting uncertain observations, W-WSD is far better than other natural-language processing context recognition functions. Moreover, W-WSD is a benchmark application in the research areas of machine learning due to its clear problem identification, altering the nature of words' senses, and high dimensionality [62]. Table 2 shows the test results of three sentiment analyzers in which we can see the sentiment assigned by each analyzer.

Function 3 listed below calculates the subjectivity and polarity of processed tweets using each sentiment analyzer (SentiWordNet, W-WSD, TextBlob) with the use of Python code.

Function 3: F3 (PT, SO, P1, P2, P3)	
Input Attributes:	**Output Attributes:**
PT: Pre-processed tweet from database	**SO:** Subjectivity/objectivity calculation of the sentence. (Objective < 0.5 ≥ Subjective)
	P1: Polarity using TextBlob [59].
	P2: Polarity using SentiWordNet [60] with unigram approach.
	P3: Polarity using wordnet with Word Sense Disambiguation (WSD) Library [63].

Function 4 below determines the sentiment, based on the subjectivity and polarity.

Function 4: F4 (SO, P1, P2, P3, S1, S2, S3)	
Input Attributes:	**Output Attributes:**
SO: Subjectivity/objectivity of each tweet.	**S1:** Sentiment of polarity calculated by TextBlob (SO ≥ 0.5 && (Positive: P1 > 0, Neutral: P1 = 0, Negative: P1 < 0)
P1: Polarity through TextBlob.	**S2:** Sentiment of polarity calculated by with unigram approach (SO ≥ 0.5 && (Positive: P2 > 0, Neutral: P2 = 0, Negative: P2 < 0)
P2: Polarity through SentiWordNet [48] with unigram approach.	**S3:** Sentiment of polarity calculated by using wordnet with Word Sense Disambiguation (WSD) Library (SO ≥ 0.5 && (Positive: P3 > 0, Neutral: P3 = 0, Negative: P3 < 0)
P3: Polarity through wordnet using Word Sense Disambiguation (WSD) Library.	

Table 2. Sentiment analysis of a tweet from three analyzers.

Sentiment Analyzer	Tweet	Sentiment Score
W-WSD	'Right move at wrong time #JIT'	Negative
TextBlob	'Right move at wrong time #JIT'	Negative
SentiWordNet	'Right move at wrong time #JIT'	Positive

E- Validation with Waikato Environment for Knowledge Analysis (Weka)

In order to validate the results obtained from TextBlob, W-WSD and SentiWordNet, we used the Waikato Environment for Knowledge Analysis (Weka). When we calculated the polarity and subjectivity of the Twitter dataset, the resultant files were in comma separated value (csv) format. To make them compatible for validation, we converted the csv files into attribute-relation file format (arff). Figure 1b illustrates the model construction and evaluation process. For model building, we applied supervised machine-learning algorithms, Naïve Bayes on the training dataset, and percentage split for SVM. Steps of the analyzers' validation through Naïve Bayes and SVM validation can be viewed from the pseudocode "Naive Bayes Classifier" and "SVM Classifier". These machine-learning algorithms (Naïve Bayes and SVM) were applied on the training set to build an analysis model. On the basis of the model constructed for each analyzer, the test set was evaluated. After test set evaluation, we recorded the accuracy of each analyzer under each model.

1. Naïve Bayes Classifier (With Training Set Scheme)

Step 1: Create data files for the classifier.

(1.1) Create a file of tweets with their sentiment of each sentiment analyzer (test set).

(1.2) Create a file of negative and positive labeled tweets of each sentiment analyzer (training set).

(1.3) Convert all csv files to arff format file for Weka compatibility.

Step 2: Build Naïve Bayes classifier model on Weka.

(2.1) Create model of each analyzer by providing the training set file.

Step 3: Execution of model on test set.

(3.1) Load the test set file.

(3.2) Apply the StringToWordVector filter with following parameters:

IDFTransform: true, TFTransform: true, stemmer: SnowballStemmer, stopwordsHandler: rainbow, tokenizer: WordTokenizer.

(3.3) Execute the model on the test set.

(3.4) Save results in the output file.

2. SVM Classifier (With Percentage Split Scheme)

Step 1: Create data files for the classifier.

(1.1) Create a file of tweets with their sentiment of each sentiment analyzer.

(1.2) Convert all csv files to arff format file for Weka compatibility.

Step 2: Execution of SVM on Twitter File.

(2.1) Load the file of tweets with their labeled sentiment.

(2.2) Apply the StringToWordVector filter with following parameters:

IDFTransform: true, TFTransform: true, stemmer: SnowballStemmer, stopwordsHandler: rainbow, tokenizer: WordTokenizer.

(2.3) Apply the percentage split approach with 70%.

(2.4) Execute and save the results in the output file.

4. Results and Discussion

This section contains the details of experiments conducted in this research along with a discussion of the results. At the first step of sentiment analysis with the aim of determining the accuracy of these analyzers, tweets were analyzed from various sentiment analyzers. The sole purpose of using a variety of sentiment analyzers such as TextBlob, SentiWordNet, and WSD was to give a comparison of their accuracies. Table 3 carries a representation of tweets analyzed from TextBlob, SentiWordNet and W-WSD with same sentiments, to a total number of 2073.

Table 3. Sentiment assignment to specific tweets for each analyzer.

Sentiment	No of Tweets
Positive	1245
Neutral	33
Negative	795

Tweets gathered from public accounts were 100,000 in number. However, after pre-processing only 6250 tweets remained. Among the three sentiment analyzers we compared in this research, we found that TextBlob had the highest rate of tweets with positive sentiment, 3380 in number and 54.08% in percentage. SentiWordNet gave 3054 the highest negative sentiment rate, 48.86%, which can be viewed in Table 4 and Figure 2.

Table 4. Polarity calculation with sentiment analyzer percentage accuracy.

Sentiment Analyzer	Positive	Neutral	Negative
TextBlob	3380 (54.08%)	667 (10.672%)	2203 (35.248%)
SentiWordNet	2593 (41.488%)	603 (9.648%)	3054 (48.864%)
W-WSD	3189 (51.024%)	1079 (17.264%)	1982 (31.712%)
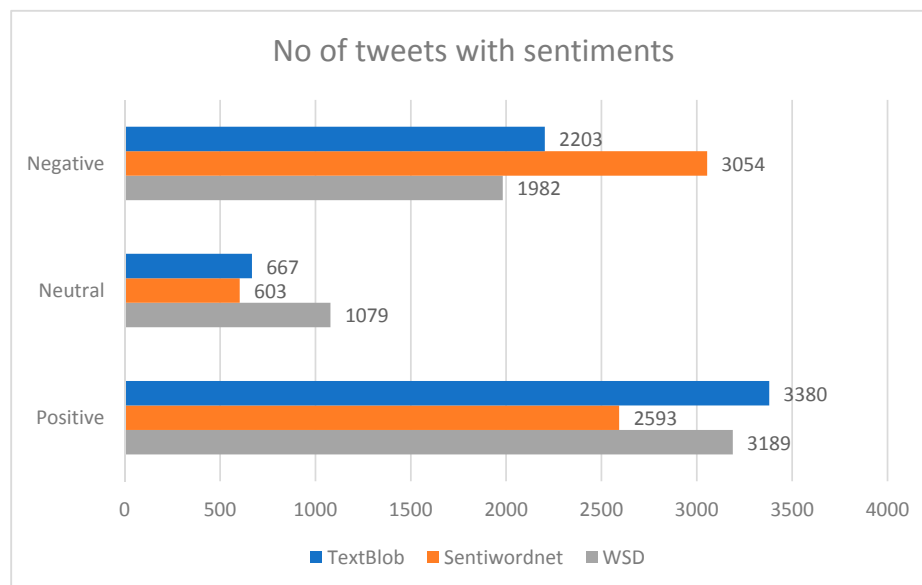
Figure 2. Polarity calculation with each sentiment analyzer.

To validate the sentiments (positive, neutral, negative) of tweets that were extracted from different sentiment analyzers, we used Weka in which we applied the Naïve Bayes classifier on a 1690 training set of 845 positive and 845 negative tweets analyzed by each sentiment analyzer. On the other hand, the test set consisted of 400 tweets. As a result, we found that W-WSD had the highest accuracy rate for predicting election sentiments when analyzed with the Naïve Bayes classifier, which gave us 316 correct instances, as shown in Table 5, and 79% accuracy, which can be viewed in Figure 3. The accuracy of TextBlob was observed to be second, with 304 correct instances and an accuracy of 76%. SentiWordNet's accuracy was relatively low in comparison to W-WSD and TextBlob, giving only 219 correct instances with 54.75%.
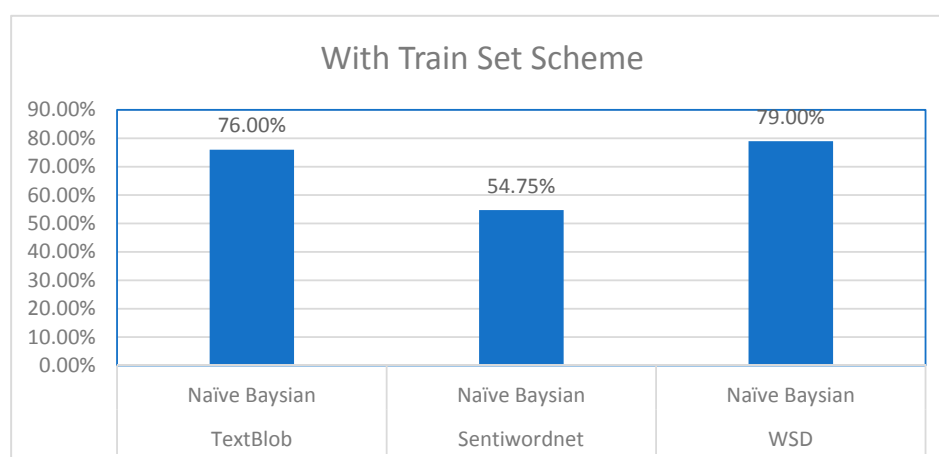

Figure 3. Accuracy obtained for each sentiment analyzer with the Naïve Bayes classifier.

Table 5. Results obtained for sentiment analyzers with Naïve Bayes.

Sentiment Analyzer	Test Set	Training Set	Accuracy	Correct Instances
TextBlob	400	1690	76.00%	304
SentiWordNet	400	1690	54.75%	219
W-WSD	400	1690	79.00%	316

Moreover, we also validated 5000 tweets by using the SVM classifier with a 70% percentage split. Figure 4 also shows that TextBlob and W-WSD have almost the same accuracy of sentiments, which was about 62%.
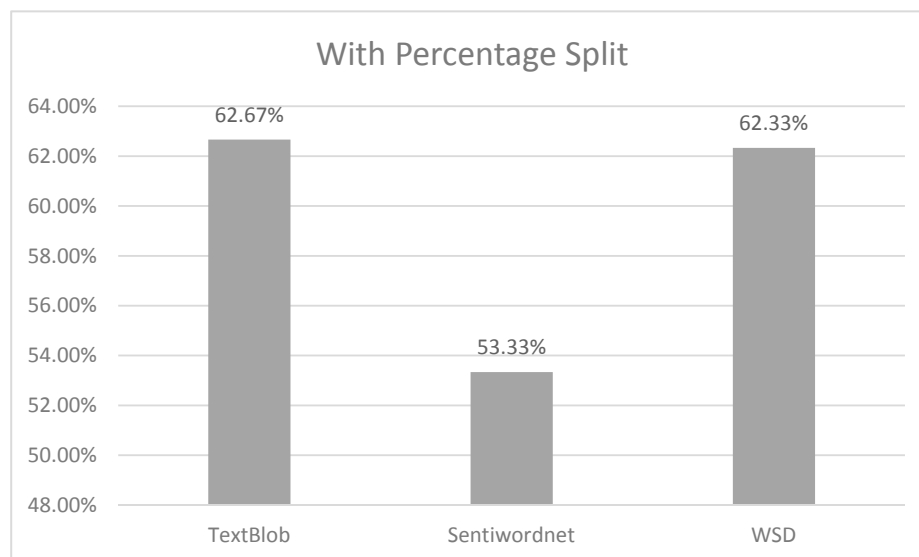
Figure 4. Results obtained from support vector machines (SVM) of each analyzer after a 70% data split.

From the experimental results presented above, TextBlob has the highest accuracy in comparison with W-WSD and TextBlob. Keeping in mind the results of our analysis, we can say that Textblob and W-WSD are far better than the SentiWordNet approach for adoption in the analysis of election sentiments and for making more accurate predictions.

5. Conclusions and Future Work

This paper focuses on the adoption of various sentiment analyzers with machine-learning algorithms to determine the approach with the highest accuracy rate for learning about election sentiments. In a lexicon-based sentiment analysis, semantic orientation is of words, phrases or sentences calculated in a document. Polarity in the lexicon-based method is calculated on the basis of the dictionary, that consists of a semantic score of a particular word. However, the approach of machine learning is basically destined to classify the text by applying algorithms such as Naïve Bayes and SVM on the files. Considerable work has been done in the field of sentiment analysis either from sentiment lexicons or from machine-learning techniques. But, this research is focused on providing a comparison between sentiment lexicons (W-WSD, SentiWordNet, TextBlob) so that the best can be adopted for sentiment analysis. We validated three of the sentiment analysis lexicons with two machine-learning algorithms, which we have not found in any previous research of this kind. As a result, we calculated sentiments from three analyzers named SentiWordNet, TextBlob, and W-WSD. Additionally, we tested their results with two supervised machine-learning classifiers, Naïve Bayes and SVM. Although the results of TextBlob were relatively better, we obtained the best result when analyzing tweets with W-WSD, as is clearly shown in the results. In order to take our initiative to next level, we will find the patterns of political parties based on Twitter reviews in future research.

This <mark>validation can be used as an on-the-fly approach for the results validation obtained from sentiment lexicons</mark>. The translation of the Urdu language is also a novel approach to this research which was not present in any previous work. In conclusion, from the comparison that has adopted a hybrid approach for sentiment analysis we have learned that TextBlob and Wordnet use word sense disambiguation with greater accuracies and can be used further in predicting elections. In addition to that, the future of sentiment analysis in this domain lies in resolving the challenges of the validation of techniques in Urdu or other languages. Tools must be able to classify/validate various language sentiments with their relevant corpus.

Author Contributions: The author S.M. contributed to the collection of hashtags from twitter profiles manually. The author A.H. code for retrieving the hashtags from Twitter profiles. A.H. and S.M. both contributed to the Tweets retrieval. A.H. did the preprocessing of tweets. A.H. and S.M. both contributed to the translation of Urdu tweets into English. A.H. assign sentiment scores to the tweets with the help of sentiment lexicons. A.H. A.K. and S.S. convinced in the experiments and result validation with Weka. S.M. wrote the paper and A.K. and S.S. checked the paper technically.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113. [CrossRef]
2. Sebastiani, F. Machine learning in automated text categorization. *ACM Comput. Surv.* **2002**, *34*, 1–47. [CrossRef]
3. Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; Stede, M. Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **2011**, *37*, 267–307. [CrossRef]
4. Prabowo, R.; Thelwall, M. Sentiment analysis: A combined approach. *J. Informetr.* **2009**, *3*, 143–157. [CrossRef]
5. Dang, Y.; Zhang, Y.; Chen, H. A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. *IEEE Intell. Syst.* **2010**, *25*, 46–53. [CrossRef]
6. Cambria, E. Affective computing and sentiment analysis. *IEEE Intell. Syst.* **2016**, *31*, 102–107. [CrossRef]
7. Jagdale, O.; Harmalkar, V.; Chavan, S.; Sharma, N. Twitter mining using R. *Int. J. Eng. Res. Adv. Tech.* **2017**, *3*, 252–256.
8. Anjaria, M.; Guddeti, R.M.R. Influence factor based opinion mining of twitter data using supervised learning. In Proceedings of the 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 6–10 January 2014; pp. 1–8.
9. Dubey, G.; Chawla, S.; Kaur, K. Social media opinion analysis for indian political diplomats. In Proceedings of the 2017 7th International Conference on Cloud Computing, Data Science & Engineering, Noida, India, 12–13 January 2017; pp. 681–686.
10. Liu, B.; Hu, M.; Cheng, J. Opinion observer: Analyzing and comparing opinions on the web. In Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, 10–14 May 2005; pp. 342–351.
11. Razzaq, M.A.; Qamar, A.M.; Bilal, H.S.M. Prediction and analysis of pakistan election 2013 based on sentiment analysis. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), Beijing, China, 17–20 August 2014; pp. 700–703.
12. Rainbow Tool. Available online: https://www.cs.cmu.edu/~mccallum/bow/rainbow/ (accessed on 26 February 2018).
13. Soucy, P.; Mineau, G.W. A simple knn algorithm for text categorization. In Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 29 November–2 December 2001; pp. 647–648.
14. Lewis, D.D. Naive (bayes) at forty: The independence assumption in information retrieval. In Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany, 21–23 April 1998; pp. 4–15.
15. Segnini, A.; Motchoffo, J.J.T. Random Forests and Text Mining. Available online: http://www.academia.edu/11059601/Random_Forest_and_Text_Mining (accessed on 26 February 2018).
16. Raschka, S. Naive bayes and text classification i-introduction and theory. *arXiv* **2014**.
17. Porter, M.F. An algorithm for suffix stripping. *Program* **1980**, *14*, 130–137. [CrossRef]
18. Willett, P. The porter stemming algorithm: Then and now. *Program* **2006**, *40*, 219–223. [CrossRef]

19.	Ramos, J. Using TF-IDF to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning, Banff, AB, Canada, 27 February–1 March 2003; pp. 133–142.

20.	Downloading and Installing Weka. Available online: http://www.cs.waikato.ac.nz/ml/weka/downloading.html (accessed on 26 February 2018).

21.	Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The weka data mining software: An update. *ACM SIGKDD Explor. Newslett.* **2009**, *11*, 10–18. [CrossRef]

22.	Miranda Filho, R.; Almeida, J.M.; Pappa, G.L. Twitter population sample bias and its impact on predictive outcomes: A case study on elections. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Paris, France, 25–28 August 2015; pp. 1254–1261.

23.	Castro, R.; Kuffó, L.; Vaca, C. Back to# 6d: Predicting venezuelan states political election results through twitter. In Proceedings of the 2017 Fourth International Conference on eDemocracy & eGovernment (ICEDEG), Quito, Ecuador, 19–21 April 2017; pp. 148–153.

24.	Kanavos, A.; Nodarakis, N.; Sioutas, S.; Tsakalidis, A.; Tsolis, D.; Tzimas, G. Large scale implementations for twitter sentiment classification. *Algorithms* **2017**, *10*, 33. [CrossRef]

25.	Kanavos, A.; Perikos, I.; Hatzilygeroudis, I.; Tsakalidis, A. Emotional community detection in social networks. *Comput. Electr. Eng.* **2017**, *65*, 449–460. [CrossRef]

26.	Jose, R.; Chooralil, V.S. Prediction of election result by enhanced sentiment analysis on twitter data using word sense disambiguation. In Proceedings of the 2015 International Conference on Control Communication & Computing India (ICCC), Trivandrum, India, 19–21 November 2015; pp. 638–641.

27.	Twitter Apps. Available online: http://www.tweepy.org/ (accessed on 26 February 2018).

28.	Esuli, A.; Sebastiani, F. *Sentiwordnet: A High-Coverage Lexical Resource for Opinion Mining*; Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR): Pisa, Italy, 2006.

29.	Miller, G.A. Wordnet: A lexical database for english. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]

30.	Hogenboom, A.; Van Iterson, P.; Heerschop, B.; Frasincar, F.; Kaymak, U. Determining negation scope and strength in sentiment analysis. In Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, USA, 9–12 October 2011; pp. 2589–2594.

31.	Navigli, R. Word sense disambiguation: A survey. *ACM Comput. Surv.* **2009**, *41*, 10. [CrossRef]

32.	Ibrahim, M.; Abdillah, O.; Wicaksono, A.F.; Adriani, M. Buzzer detection and sentiment analysis for predicting presidential election results in a twitter nation. In Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA, 14–17 November 2015; pp. 1348–1353.

33.	Kaggle. Mean Absolute Error. Available online: https://www.kaggle.com/wiki/MeanAbsoluteError (accessed on 26 February 2018).

34.	Dokoohaki, N.; Zikou, F.; Gillblad, D.; Matskin, M. Predicting swedish elections with twitter: A case for stochastic link structure analysis. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Paris, France, 25–28 August 2015; pp. 1269–1276.

35.	Liben-Nowell, D.; Kleinberg, J. The link-prediction problem for social networks. *J. Assoc. Inf. Sci. Technol.* **2007**, *58*, 1019–1031. [CrossRef]

36.	Foreman, E. *Survey Sampling Principles*; CRC Press: Boca Raton, FL, USA, 1991.

37.	Damaschin, M.; Dorst, M.; Gerontini, M.; Imamoglu, C.; Queva, C. Reputation Management System. U.S. Patent 8,112,515, 7 Feburary 2012.

38.	Rezapour, R.; Wang, L.; Abdar, O.; Diesner, J. Identifying the overlap between election result and candidates' ranking based on hashtag-enhanced, lexicon-based sentiment analysis. In Proceedings of the 2017 IEEE 11th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 30 January–1 February 2017; pp. 93–96.

39.	Ding, X.; Liu, B.; Yu, P.S. A holistic lexicon-based approach to opinion mining. In Proceedings of the 2008 International Conference on Web Search and Data Mining, Palo Alto, CA, USA, 11–12 February 2008; pp. 231–240.

40.	Pennebaker, J.W.; Francis, M.E.; Booth, R.J. *Linguistic Inquiry and Word Count: Liwc 2001*; Mahway: Lawrence Erlbaum Associates: Mahwah, NJ, USA, 2001; Volume 71.

41. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up: Sentiment classification using machine learning techniques. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, USA, 6–7 July 2002; Volume 10, pp. 79–86.
42. Gautam, G.; Yadav, D. Sentiment analysis of twitter data using machine learning approaches and semantic analysis. In Proceedings of the 2014 Seventh International Conference on Contemporary Computing (IC3), Noida, India, 7–9 August 2014; pp. 437–442.
43. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany, 21–23 April 1998; pp. 137–142.
44. Berger, A.L.; Pietra, V.J.D.; Pietra, S.A.D. A maximum entropy approach to natural language processing. *Comput. Linguist.* **1996**, *22*, 39–71.
45. Conover, M.D.; Gonçalves, B.; Ratkiewicz, J.; Flammini, A.; Menczer, F. Predicting the political alignment of twitter users. In Proceedings of the 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, Boston, MA, USA, 9–11 October 2011; pp. 192–199.
46. Le, H.; Boynton, G.; Mejova, Y.; Shafiq, Z.; Srinivasan, P. Bumps and bruises: Mining presidential campaign announcements on twitter. In Proceedings of the 28th ACM Conference on Hypertext and Social Media, Prague, Czech Republic, 4–7 July 2017.
47. Developers. Twitter Rest API. Available online: https://dev.twitter.com/rest/public (accessed on 26 February 2018).
48. Baccianella, S.; Esuli, A.; Sebastiani, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Proceedings of the International Conference on Language Resources and Evaluation, Valletta, Malta, 17–23 May 2010; pp. 2200–2204.
49. Khan, F.H.; Bashir, S.; Qamar, U. Tom: Twitter opinion mining framework using hybrid classification scheme. *Decis. Support Syst.* **2014**, *57*, 245–257. [CrossRef]
50. Khatua, A.; Khatua, A.; Ghosh, K.; Chaki, N. Can# twitter_trends predict election results? Evidence from 2014 indian general election. In Proceedings of the 48th Hawaii International Conference on System Sciences, Kauai, HI, USA, 5–8 January 2015; pp. 1676–1685.
51. Hansen, L.K.; Arvidsson, A.; Nielsen, F.Å.; Colleoni, E.; Etter, M. Good friends, bad news-affect and virality in twitter. In *Future Information Technology*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 34–43.
52. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 168–177.
53. Saha, S.; Yadav, J.; Ranjan, P. Proposed approach for sarcasm detection in twitter. *Indian J. Sci. Technol.* **2017**, *10*. [CrossRef]
54. Turney, P.D. Thumbs up or thumbs down: Semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 417–424.
55. Sharma, P.; Moh, T.-S. Prediction of indian election using sentiment analysis on hindi twitter. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 1966–1971.
56. Wang, M.; Shi, H. Research on sentiment analysis technology and polarity computation of sentiment words. In Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing, Shanghai, China, 10–12 December 2010; pp. 331–334.
57. Roesslein, J. Tweepy Documentation. 2009. Available online: http://docs.tweepy.org/en/v3.5.0/ (accessed on 26 February 2018).
58. MTRanslator. Available online: https://github.com/mouuff/mtranslate (accessed on 26 February 2018).
59. Loria, S.; Keen, P.; Honnibal, M.; Yankovsky, R.; Karesh, D.; Dempsey, E. Textblob: Simplified Text Processing. Available online: https://textblob.readthedocs.org/en/dev/ (accessed on 26 February 2018).
60. SentiWordNet. Available online: http://sentiwordnet.isti.cnr.it/ (accessed on 26 February 2018).

61. Farooq, U.; Dhamala, T.P.; Nongaillard, A.; Ouzrout, Y.; Qadir, M.A. A word sense disambiguation method for feature level sentiment analysis. In Proceedings of the 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Kathmandu, Nepal, 15–17 December 2015; pp. 1–8.

62. Agirre, E.; Edmonds, P. *Word Sense Disambiguation: Algorithms and Applications*; Springer Science & Business Media: Berlin, Germany, 2007; Volume 33.

63. Sentiment Classifier. Available online: https://github.com/kevincobain2000/sentiment_classifier (accessed on 26 February 2018).