Comparative Study of Machine Learning Algorithms for Social Media Text Analysis

Chapter · May 2020

DOI: 10.1007/978-981-15-5830-6_19

2 authors, including:

Saksham Jain
Guru Gobind Singh Indraprastha University
5 PUBLICATIONS **92** CITATIONS

SEE PROFILE

Comparative Study of Machine Learning Algorithms for Social Media Text Analysis

Nidhi Malik(✉) and Saksham Jain

Amity School of Engineering and Technology, New Delhi, India
nidhimalikl4@gmail.com, sakshamjn655@gmail.com

Abstract. This paper highlights the way different machine learning algorithms are used in analyzing social media text. This is the internet age. People make use of online forums, blogs posts, tweets etc. to communicate with each other. As a result of increased social networking, amount of data generated is enormous. This data is an excellent source of information in all walks of life ranging from business, marketing, trends analysis and prediction etc. Sentiment analysis refers to identification of user-generated text as positive or negative or neutral automatically. This classification of sentiments into classes can be done based on the document, Sentence, Feature or Aspect. This paper presents how machine learning techniques are used for analyzing sentiments expressed on Twitter platform. Comparative study of these machine learning techniques is done for better understanding.

Keywords: Machine learning · Twitter · Sentiment analysis

1 Introduction

The ever growing usage of social media has raised the need of analyzing its text. It is an excellent source of information in all walks of life ranging from business, marketing, trends analysis and prediction etc. According to the Oxford dictionary [1], sentiment analysis is termed as classifying opinions expressed through text to decide whether the opinion about a specific topic or product is positive or negative or neutral. Opinion Mining also stands for Twitter Sentiment Analysis, on dataset taken from Twitter, is primarily for analyzing opinions, views and conversations for various purposes like deciding the business strategy to get feedback for products, political analysis, and also for assessing public actions. Analysis of sentiment is a complex process involving 5 different steps to analyze the polarity of data. These steps are: a) Data collection, b) Text pre-processing, c) training classifier, d) Sentiment classification and e) Evaluation of Output.

Initially, it was mainly used to analyze sentiment based on long texts such as emails, letters, passages and so on. Python and R are mostly used for analysis of sentiment of a particular dataset. Machine learning approaches are used for predicting the polarity of sentiments based upon training examples and testing data-sets. Natural Language Processing, algorithms like SVM, Logistics regression, Naïve Bayes etc. are used for predicting the polarity of the sentence. There are many others factors on which sentiments may depend such as aspect, sentence, and document. It is not sufficient only to use positive and negative words for analyzing sentiment of the sentence, rather context is very significant in determining the sentiment of the text block.

U. Batra et al. (Eds.): REDSET 2019, CCIS 1230, pp. 223–235, 2020.
https://doi.org/10.1007/978-981-15-5830-6_19

The paper is organized into six sections. First section introduces sentiment analysis and its need. Second section highlights the related work done and their features. In third section, we have discussed different approaches that can be used for sentiment analysis. Fourth section describes various machine learning techniques used for analyzing sentiments. In fifth section, we have evaluated different techniques and also analyzed their performance. The paper is concluded in sixth section.

2 Related Work

With the huge increase in number of Blogging and social networking sites and their user-base, sentiment and opinion analysis are becoming a new field/domain of interest for many academic-researchers as well as industry experts. Different sentimental analysis techniques have been implemented by researchers on Twitter data. For example, the work done by [2] is a real time twitter sentiment analysis for presidential elections. They have followed an approach based on crowdsourcing for sentiment annotation on in-domain political data. This data is then used for generating data for model training and testing. [3] have used labeled corpora and explained a method which automatically builds a lexicon which consists of extremely negative and positive words from this labeled corpus. Extreme reviews are then searched from the classifier which classifies based on the labeled corpora. [4] have used the IMDb dataset which contains movie reviews to further use machine learning algorithms for classification techniques using, Unigram, Bigram, Trigram, combination of unigram & bigram, bigram & trigram, and unigram & bigram & trigram. [6] discuss polarity of lexicon based sentiment analysis and semantic-orientation of words, phases, and sentences on the basis of dictionary consisting of semantic scores. And comparison of W-WSD, SentiWordNet, and Text Blob lexicon models. SMS spam filtering has been proposed by [7] by proposing a pre-processing approach for normalizing along with enhancing its performance of classifying SMS's. Text-blob is a python library for processing words which uses NLTK (natural language toolkit) for natural language processing. Large and structured set of texts known as corpora which is used for the analysis of tweets as illustrated in [12]. A novel solution to target oriented (aspect based) sentiment summarization and SA of short informal texts is described in [9]. [8] use the model for classification of tweets by using 3 approaches: Sentic-Net, Senti-WordNet and Sentis-LangNet [9]. Presents a solution to target oriented sentiment summarization as well as sentiment analysis of short informal texts mainly tweets.

3 Approaches for Sentiment Classification

While going through literature, we observed that there are two widely used approaches for detecting sentiment from text. Lexical based and Machine learning based approaches are the two main approaches used by researchers.

Lexical based approach is also termed as dictionary based approach and it relies on the use of lexicon and dictionary of words. Its performance depends on the lexical on which it is based. Techniques such as word sense disambiguation etc. can be used to enhance performance of lexical based approaches.

As discussed by [14], Machine learning is sub-divided into 2 categories- Supervised learning and Unsupervised learning. Supervised learning algorithms are probabilistic, linear, decision tree and rule based classifier, in which labeled dataset is provided as an input for training the classifier and thus model is tested on new data to get outputs. Sentiment classification in ml includes 2 steps. First includes extracting features and storing feature vector and second is to train feature vector by using classification algorithms whereas in case of Unsupervised learning, there is no category and they are not provided with the correct targets so it is dependent on the process of clustering. One of the major things to take care of while training the model is Overfitting which means that the classifier works well on your training data but on introducing it to new data its accuracy decreases or we can say it acts poorly on new data that's why machine learning experts train the models on different data then the data used of knowing the accuracy of the classifier.

4 Techniques Used

We test different classifiers: keyword-based, Naive Bayes, Logistics regression, support vector machines, random forest and finally K-nearest neighbor algorithms.

Data Description

Twitter consist of short messages, with an upper limit of 140 characters recently changed to 280 words. In micro-blogging services people often use quick and short messages, acronyms, make spelling mistakes, use emoticons along with characters that have unique meanings. Following is a brief terminology associated with tweets. For the purpose of experiments to be conducted in this paper, we have used a dataset named "Sentiment 140" which includes roughly about 1.6 million tweets labeled "1" for a positive response and "0" for a negative response. From this dataset, we need only two variables which are "Sentiment" and "SentimentText" containing polarity of tweets and the tweets respectively (Fig. 1).

A	B	C	D
ItemID	Sentiment	SentimentSource	SentimentText
1	0	Sentiment140	is so sad for my APL friend.....
2	0	Sentiment140	I missed the New Moon trailer...
3	1	Sentiment140	omg its already 7:30 :O
4	0	Sentiment140	.. Omgaga. Im sooo im gunna CRy. I've been at this dentist since 11.. I was suposed 2 j
5	0	Sentiment140	i think mi bf is cheating on me!!! T_T
6	0	Sentiment140	or i just worry too much?
7	1	Sentiment140	Juuuuuuuuuuuuuuussssst Chillin!!
8	0	Sentiment140	Sunny Again Work Tomorrow :-| TV Tonight
9	1	Sentiment140	handed in my uniform today . i miss you already
10	1	Sentiment140	hmmmm.... i wonder how she my number @-)
11	0	Sentiment140	I must think about positive..
12	1	Sentiment140	thanks to all the haters up in my face all day! 112-102
13	0	Sentiment140	this weekend has sucked so far
14	0	Sentiment140	jb isnt showing in australia any more!
15	0	Sentiment140	ok thats it you win.

Fig. 1 Attributes of our dataset.

Data Cleaning (Pre-processing)

1) Filtering – we remove URL links (for e.g. https://Sakshamjain.me)
2) Replaces targets (e.g. "@John") with blank spaces
3) Tokenization – In tokenization long sentences are converted by using space between adjoining words and by using punctuation marks to make a collection of words called as bag of words. However, ensuring that short forms such like "don't", "I'll", "she'd" remained as a single word.
4) Stopwords– By removing articles like ("a", "an", "the", " ", "[", "{", etc.) from collection of words.
5) N-grams – By making a set of words out of adjacent words. For ex: In a Trigram system, sentence like "I do not like fish" will form three bi-grams: "I + do + not", "do + not like", "not + like fish".

Bag of Words Models

Ngrams are surprisingly powerful given their simplicity, they are fast to train and easy to understand. Even though ngram's bring some context between words, a bag of word models fails in modeling long-term dependencies between words in a sequence.

1) Word ngram: Word N-grams are described as combinations of adjacent words of length n each from the text present as the input function. we have used unigrams (n = 1) and bigrams (n = 2) as features for our classification.
2) Character ngram:
   It is very similar to word ngram but instead of words, a group of character is implemented here. We used a range of 1 to 4 for getting a list of possible occurrences of the instances.
3) Word-character ngram
   Basically, it is a combination of word n-gram and character ngram thus combining the list of all possible entities of word and character Ngrams. This is done so as to combine the features of each of them for getting higher accuracy by our classifier.

Evaluation of the Models

Performance metrices solely depends upon the type of problem being evaluated. Confusion matrix measures the performance such that output can be two or more classes. A table is generated which gives different combinations of predicted and actual values [13]. For better interpretation of the table, we can also see this in terms of true positives, true negative, false positive, and false negatives. Following are different parameters:

Accuracy

$$Accuracy = \frac{true\ positives + true\ negatives}{total\ examples} \tag{1}$$

Accuracy determines the correctness of training of the model and its behavior in real world. Although it will not provide any detailed information about it's application for the problem. Accuracy just tell us about the efficacy of the model trained.

Precision

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{2}$$

Precision helps in determining whether the occurrence of False positives is higher than True positives. For example, in case of detecting lung cancer for an individual after analysing their test records. If model under consideration has less precision then results of the analysis can't be trusted upon as it might be a false positive. And if ration of false positives is higher than True positive then further analysis is needed to confirm the results.

Recall

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{3}$$

Recall helps when there are higher number of false negatives. When the frequency of false negatives is higher than it degrades the efficiency of our model. For example: news forecast is for rain and although seeing it you decide to carry an umbrella but incident didn't happened and later you thinking it was a bad forecast.

F1-Score

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

F1 score helps in determining the overall measure of the classifier's accuracy. F1 score combines precision and recall both. A good F1 score means that probability of both false positives and false negatives is less, therefore we can say the we can correctly classify threats in real world scenarios.

4.1 Techniques Used

We've tested different classifiers: keyword-based, Naive Bayes, Logistics regression, support vector machines, random forest and finally K-nearest neighbor algorithms.

Naive Bayes

Bayes Rule is the basis for Bayesian classifiers. Its principle is to look at conditional probabilities in a way which allows to flip the condition around in a suitable way. Conditional probability is defined as a probability that event X will occur, given the evidence Y and is normally written P(X | Y). The rearrangement of finding probability of something is based on examples of it occurring can be very useful. For example, here, given the contents, we are trying to find out the probability of a document being positive or negative. Examples of positive and negative opinions from our data set are already given so it is convenient [5]. Results of our prediction for Naive bayes model is shown in Table 1 through which we get to know that best results of naive bayes are with Word-Char ngram with an accuracy of 0.6694. And graphical representation can be seen in Fig. 2 in the form of bar-chart.

Table 1. Results of Naive Bayes model.

Type	Accuracy	Precision	Recall	F1 score
Word	0.6540	0.5334	0.7098	0.6091
Char	0.6614	0.6981	0.6548	0.6757
Word-Char	0.6694	0.6680	0.6747	0.6713

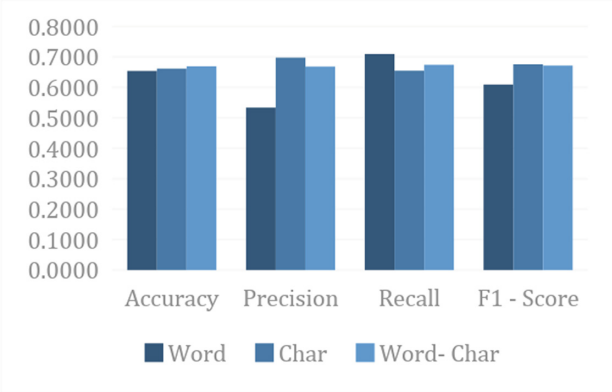

Fig. 2. Representation of Naive Bayes results

Support Vector Machine

The support vector machine algorithm is based on the principle of structural risk minimization, rather than the traditional empirical risk minimization principle. In SVM, based on the training dataset, the decision boundaries are directly determined for which the separating margins of the boundaries can be maximized in feature space. Results of our prediction for SVM model is shown in Table 2 which shows that we get best results of an accuracy of 0.7710 for Word-Char n-gram. Graphical representation can be seen in Fig. 3 in the form of bar-chart.

A separating hyperplane is written as:

$$W * X + b = 0$$

$$K(\mathbf{X}_i, \mathbf{X}_j) = \begin{Bmatrix} \mathbf{X}_i.\mathbf{X}_j & \text{Linear} \\ (\gamma \mathbf{X}_i.\mathbf{X}_j + C)^d & \text{Polynomial} \\ \exp(-\gamma |\mathbf{X}_i - \mathbf{X}_j|^2) & \text{RBF} \\ \tanh(\gamma \mathbf{X}_i.\mathbf{X}_j + C) & \text{Sigmoid} \end{Bmatrix} \tag{5}$$

Table 2. Results of Support vector machine model.

Type	Accuracy	Precision	Recall	F1 score
Word	0.7480	0.7305	0.7612	0.7456
Char	0.7692	0.7713	0.7719	0.7716
Word-Char	0.7710	0.7736	0.7733	0.7735

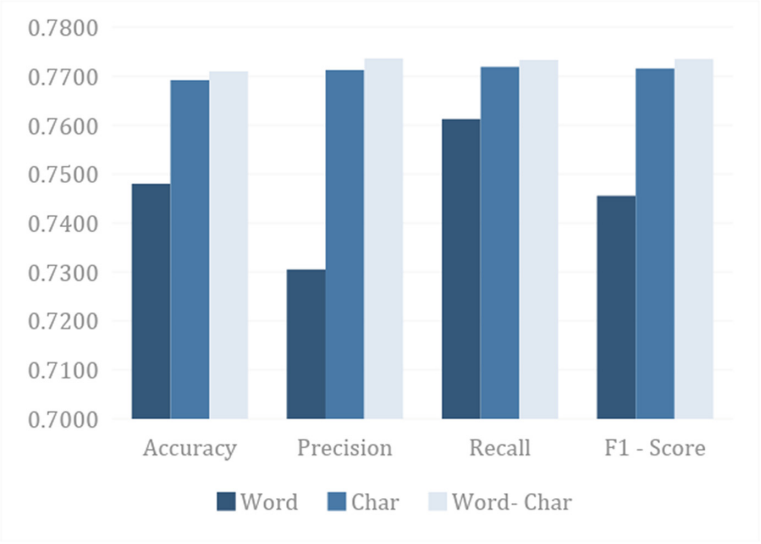

Fig. 3. Representation of SVM results

k-Nearest Neighbor

K-NN is a type of machine learning algorithm which uses feature similarity for computation of new data points. Values are assigned to the data points based on how closely it matches the points in the training set. The class assigned to an object is the most common among its k-nearest neighbours [11]. Results of our prediction for KNN model is shown in Table 3 with maximum accuracy of 0.6024 and with maximum accuracy of 0.6176. And graphical representation can be seen in Fig. 4 in the form of bar-chart.

Table 3. Results of K-NN model.

Type	Accuracy	Precision	Recall	F1 score
Word	0.6024	0.4745	0.6450	0.5467
Char	0.6118	0.6047	0.6186	0.6116
Word-Char	0.6176	0.6581	0.6450	0.6515

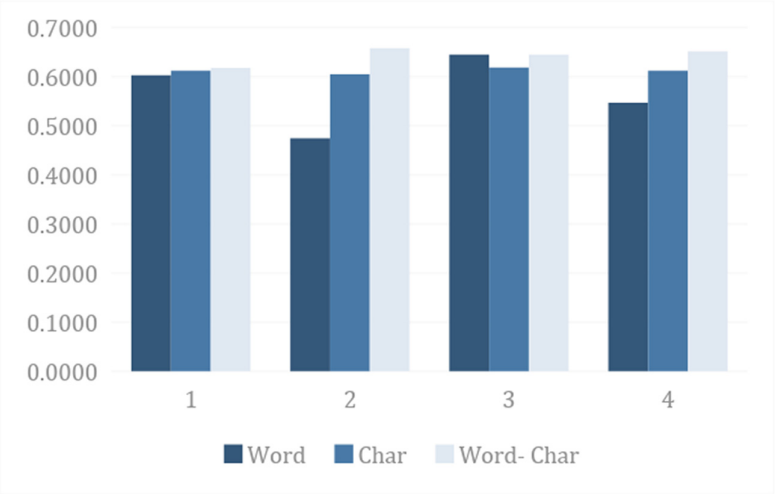

Fig. 4. Representation of K-NN results

Logistics Regression

Logistic regression is used to develop a regression model which uses logistic function to model a binary dependent variable [10]. It can be of three types (1) binary, with two possible outcomes (2) multinomial – for more than two non-ordered categories of the dependent variable and (3) ordinal – for ordered categories. Results of our prediction for Logistics Regression model is shown in Table 4 through which we can say that the maximum accuracy of approx 80%, 79.02 to be precise. And graphical representation can be seen in Fig. 5 in the form of bar-chart.

For r independent variables x1, x2 and x3.. the logistic function is:

$$\hat{p} = \frac{\exp(b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p)}{1 + \exp(b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p)} \tag{6}$$

where p is the probability of the event.

The goal is to obtain bi where i = 0, 1, 2..

Table 4. Results of Logistic Regression model.

Type	Accuracy	Precision	Recall	F1 score
Word	0.7554	0.8678	0.7116	0.7820
Char	0.7734	0.8615	0.7355	0.7935
Word-Char	0.7902	0.7891	0.7944	0.7917

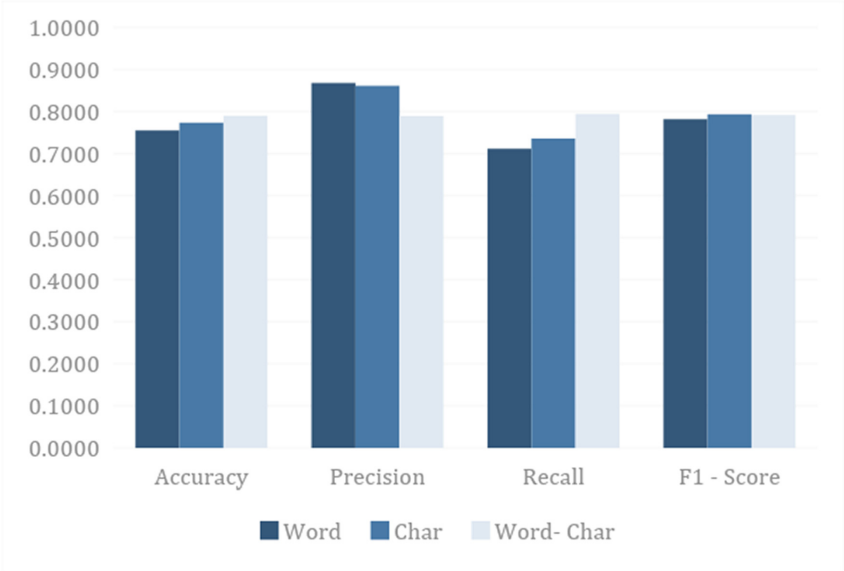

Fig. 5. Representation of Logistics Regression results

Random Forest

Random forests are constructed using the predictions of several trees. Each of these trees is trained in isolation. During Boosting, after training the base model are combined using a sophisticated weighting scheme. In this process, the trees are trained independently and the averaging is used for predictions of the trees. For construction of the random trees, there are some choices such as method for splitting the leafs, deciding which predictor should be used in each leaf and how to inject randomness into the trees [15]. Results of our prediction for Random Forest model is shown in Table 5 with an accuracy of around 73.44% with a minimum accuracy of 71.78 percentage. And graphical representation can be seen in Fig. 6 in the form of bar-chart.

$$K_k^{cc}(\mathbf{x},\mathbf{z}) = \sum_{k_1,\ldots k_d,\sum_{j=1}^{d} k_j=k} \frac{k!}{k_1!\cdots k_d!}\left(\frac{1}{d}\right)^k \prod_{j=1}^{d} \mathbf{1}_{\lceil 2^{k_j} x_j\rceil = \lceil 2^{k_j} z_j\rceil}, \quad \text{for all } \mathbf{x},\mathbf{z} \in [0,1]^d. \tag{7}$$

Table 5. Results of Random Forest model.

Type	Accuracy	Precision	Recall	F1 score
Word	0.7178	0.7990	0.6910	0.7411
Char	0.7344	0.7831	0.7173	0.7488
Word-Char	0.7344	0.8627	0.6897	0.7665

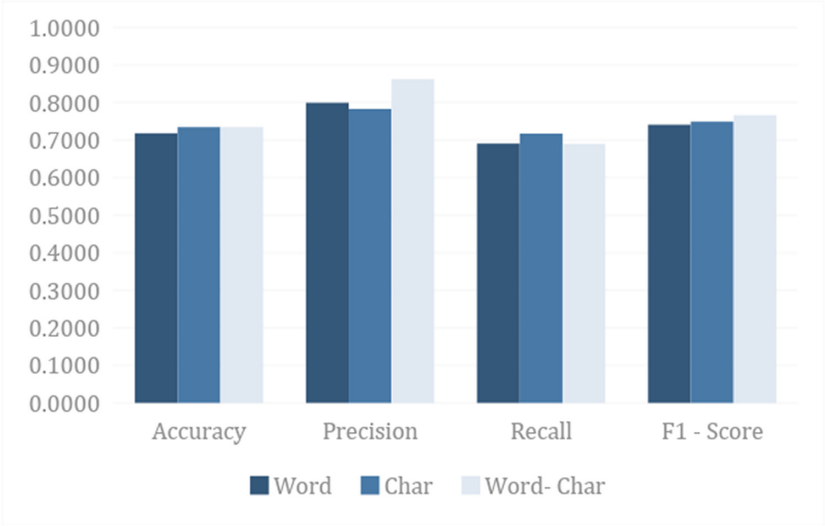

Fig. 6. Representation of Random Forest results

5 Evaluation

For word N-gram, we have validated 10000 tweets as test data and 40000 tweets as training data for different classifiers and we got an accuracy of 75.54% in Logistics regression, 65.40% in Naïve bayes, 74.80% in SVM, 71.78% in Random forest, and 60.24% in KNN.

For Char N-gram, we have validated 10000 tweets as test data and 40000 tweets as training data for different classifiers and we got an accuracy of 77.34% in Logistics regression, 66.14% in Naïve bayes, 76.86% in SVM, 72.30% in Random forest, and 61.18% in KNN.

For Word-Char N-gram, we have validated 10000 tweets as test data and 40000 tweets as training data for different classifiers and we got an accuracy of 79.02% in Logistics regression which is best amongst all of them, 66.94% in Naïve bayes, 77.10% in SVM, 73.44% in Random forest, and 61.76% in KNN.

Analysis of the prediction of tweets by our models based upon the four majorly used standards i.e. True Positive, True Negative, False Positive, and False Negative as shown in the Table 6, which depicts the comparison of different techniques- Word, Char, and Word-Char N-grams consisting of various algorithms of Machine Learning. Along with graphical representation shown in Fig. 7, the results we got by using various machine learning models corresponding with type of N-gram it was evaluated with.

Table 6. Results of Different Algorithms in terms of TP, TN, FP, FN.

Models	Type	True positive	True negative	False positive	False negative
KNN	Word	1199	1813	1328	660
	Char	1528	1531	999	942
	Word- Char	1663	1425	864	1048
Naive Bayes	Word	1348	1922	1179	551
	Char	1764	1543	763	930
	Word- Char	1688	1659	839	814
Random Forest	Word	2019	1570	508	903
	Char	1979	1693	548	780
	Word- Char	2180	1492	347	981
SVM	Word	1846	1894	681	579
	Char	1949	1897	578	576
	Word- Char	1955	1900	572	573
Logistics Regression	Word	2193	1584	334	889
	Char	2177	1690	350	783
	Word- Char	1994	1957	533	516

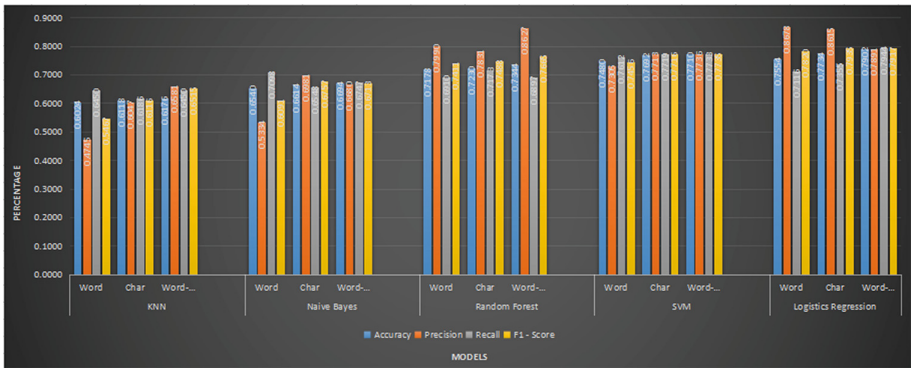

Fig. 7. Representation of Different Algorithms in terms of TP, TN, FP, FN.

Comparison of all the models along with different types of N-grams with respect to Accuracy, Precision, Recall, F1-Score also known as the four evaluation criteria is shown in Table 7. Followed by a graphical representation shown in Fig. 8, the results we got on predictions for getting the idea of which of the model works best and the model which didn't perform well.

Table 7. Results of Different Algorithms in terms of Confusion Matrix.

Models	Type	Accuracy	Precision	Recall	F1-score
KNN	Word	0.6024	0.4745	0.6450	0.5467
	Char	0.6118	0.6047	0.6186	0.6116
	Word- Char	0.6176	0.6581	0.6450	0.6515
Naive Bayes	Word	0.6540	0.5334	0.7098	0.6091
	Char	0.6614	0.6981	0.6548	0.6757
	Word- Char	0.6694	0.6680	0.6747	0.6713
Random Forest	Word	0.7178	0.7990	0.6910	0.7411
	Char	0.7344	0.7831	0.7173	0.7488
	Word- Char	0.7344	0.8627	0.6897	0.7665
SVM	Word	0.7480	0.7305	0.7612	0.7456
	Char	0.7692	0.7713	0.7719	0.7716
	Word- Char	0.7710	0.7736	0.7733	0.7735
Logistics Regression	Word	0.7554	0.8678	0.7116	0.7820
	Char	0.7734	0.8615	0.7355	0.7935
	Word- Char	0.7902	0.7891	0.7944	0.7917

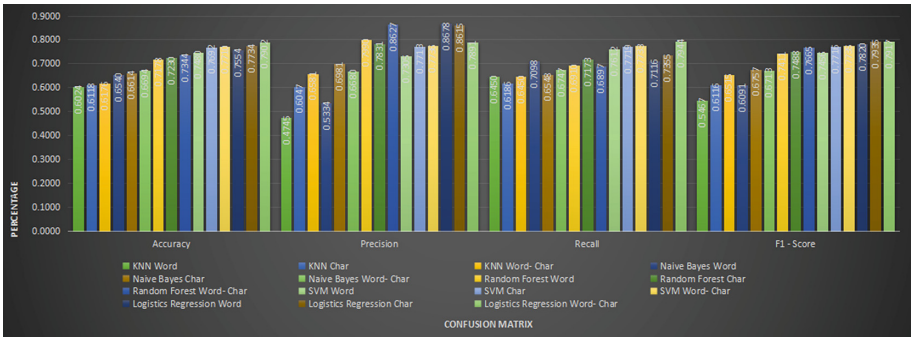

Fig. 8. Representation of Different Algorithms in terms of Confusion Matrix.

6 Conclusion

Sentiment Analysis from social media text an excellent source of information in all walks of life ranging from business, marketing, trends analysis and prediction etc. This paper presents how machine learning techniques are used for analyzing sentiments expressed on Twitter platform. In this work, upon considering F1 score we conclude that for classifying tweets using machine learning, the model word-char ngram technique combines perfectly with the model. Word-char ngram combination with our models result in getting the highest accuracy amongst all. Comparative study of all the techniques is depicted with actual results and graphical representation also for clear understanding.

References

1. https://www.linkedin.com/pulse/importance-sentiment-analysis-social-media-christine-day
2. Wang, H., Dogan, C., Kazemzadeh, A., Bar, F.: A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In: Proceedings of the ACL 2012 System Demonstrations, pp. 115–120 (2012)
3. Almatarneh, S., Gamallo, P.: A lexicon-based method to search for extreme opinions. PLoS ONE **13**, e0197816 (2018)
4. Tripathy, A., Agrawal, A., Rath, S.K.: Classification of sentiment reviews using n-gram machine learning approach. Expert Syst. Appl. **57**, 117–126 (2016)
5. Troussas, C, Virvou, M.: Sentiment analysis of facebook statuses using naive bayes classifier for language learning (2013)
6. Hasan, A., Moin, S.: Machine learning based sentiment analysis for twitter accounts. Math. Comput. Appl. **23**, 11 (2018)
7. Almeida T.A, Pontes e Silva T.B.: Text normalization and semantic indexing to enhance SMS spam filtering (2016)
8. Pandarachalil, R., Sendhilkumar, S., Mahalakshmi, G.S.: Twitter sentiment analysis for large-scale data: an unsupervised approach. Cogn. Comput. **7**(2), 254–262 (2014). https://doi.org/10.1007/s12559-014-9310-z
9. Bahrainian, S., Dengel, A.: Sentiment analysis and summarization of twitter data. In: 2013 IEEE 16th International Conference on Computational Science and Engineering, Sydney, NSW, pp. 227–234 (2013)
10. Zekic-Susan, M., Salija, N.: Predicting company growth using logistic regression and neural networks. Croatian Oper. Res. Rev. **7**, 229–248 (2016)
11. Dey, L., Chakraborty, S.: Sentiment analysis of review datasets using naïve bayes and k-nn classifier (2016)
12. TextBlob (2017). https://textblob.readthedocs.io/en/dev/
13. Skymind. https://skymind.ai/wiki/accuracy-precision-recall-f1
14. Vimalkumar, B., Vaghela, B., Jadav, M.: Analysis of various sentiment classification techniques. IJCA **140**, 975–8887 (2016)
15. Denil, M., Matheson, D.: Narrowing the gap: random forests in theory and in practice (2014)