YOUTUBE COMMENTS SENTIMENT ANALYSIS

Ritika Singh[#1], Ayushka Tiwari [*2]

1.Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Ghaziabad
2.Department of CSE, SRM Institute of Science and Technology, Ghaziabad

ABSTRACT

Over time, textual information has increased exponentially, resulting to the potential research within the field of machine learning (ML) and natural language processing (NLP). Sentiment analysis of you-tube comments is a very interesting topic nowadays.While many of these videos have a significant number of user comments and reviews, little work has been done so far in extracting trends from these comments due to their low information consistency and quality. In this paper we perform sentiment analysis on the YouTube comments related to popular topics using machine learning techniques/algorithms. We demonstrate that an analysis of the sentiments to spot their trends, seasonality and forecasts can provide a transparent picture of the influence of real-world events on public sentiments. Results show that the trends in users' sentiments are well correlated to the real-world events associated with the respective keywords. The main purpose of this research is to facilitate researchers to identify quality research papers on their sentiment analysis. In this research, sentiment analysis of you-tube comments using citation sentences is carried out using an existing constructed annotated corpus. This corpus is consisted of 1500 citation sentences. The noise was cleaned from data using different data normalization rules in order to clean the comments from the corpus. To perform classification on this data set we developed a system in which six different machine learning algorithms including Naïve-Bayes (NB), Support Vector Machine (SVM),Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbor (KNN) and Random Forest (RF) are implemented. Then the accuracy of the system is evaluated using different evaluation metrics e.g. F-score and Accuracy score.

Keywords— Sentimental analysis; citations; machine learning; classification;

1. INTRODUCTION

In this work, we will collect the data from the you-tube comments of the public and measures the attitude of the user towards the aspects of a video which they describe in a text.

Sentiment analysis is useful for quickly gaining the whole idea by using large number of text data and it will be helpful to understand the user's opinion. Sentimental analysis is additionally referred as opinion mining that means to find out or identify the positive, negative, neutral opinions, views, attitudes, impressions, emotions and feelings indicated in the text.

Current YouTube usage statistics indicate the approximate scale of the site: at the time of this writing there are quite 1 billion unique users viewing video content, watching over 6 billion hours of video each month. Also, YouTube accounts for 20% of web traffic and 10% of total internet traffic.

YouTube provides many social mechanisms to gauge user opinion and views a few video by means of voting, rating, favourites, sharing and negative comments, etc. It's important to notice that YouTube provides more than just video sharing; beyond uploading and viewing videos, users can subscribe video channels and may interact with other users through comments. YouTube is generally a comprise

of implicit and explicit user-user interaction. This user-to-user social aspect of YouTube (the YouTube social network) has been cited together key differentiating factor compared to other traditional content providers.
Text analytics is that the analysis of "unstructured" data contained in natural language text using various methods machine learning tools, and techniques. Text analysis offers a very low-cost method to gauge public opinion.

In this research work, we have done sentimental analysis of public comments by using an annotated corpus consists of citation sentences. The corpus is made up of 1500 citation sentences. The corpus is annotated using some rules to assign the polarity to citation sentences.

We've developed a system based on six different machine learning algorithms including Naïve-Bayes, Support Vector Machine, Logistic Regression, Decision Tree, KNearestNeighbor and Random Forest.Accuracy of the classification algorithms has been evaluated using different evaluations measures e.g., F-Score and Accuracy score to evaluate the classification system' correctness. To improve our system' performance, we've used different features selection techniques like lemmatization, n-gaming, tokenization, stop words and punctuation removal.

2. RELATED WORK

Several researchers have performed sentiment analysis of social networks like Twitter and YouTube .These works affect comments, tweets and other metadata collected from the social network profiles of users or of public events that are collected and analyzed to get significant and interesting insights about the usage of these social network websites by the overall mass of individuals . The work most closely associated with ours is by Siersdorfer et al. They analyzed quite 6 million comments collected from 67,000 YouTube videos to identify the connection between comments, views, comment ratings and topic categories. The authors show promising leads to predicting the comment ratings of latest unrated comments by building prediction models using the already rated comments. Pang, Lee and Vaithyanathan perform sentiment analysis on 2053 movie reviews collected from the web Movie Database (IMDb). They examined the hypothesis that sentiment analysis are often treated as a special case of topic-based text classification. Their work depicted that standard machine learning techniques such as Naive Bayes or Support Vector Machines (SVMs) outperform manual classification techniques that involve human intervention.

However, the accuracy of sentiment classification falls in need of the accuracy of ordinary topic-based text categorization that uses such machine learning techniques. They reported that the simultaneous presence of positive and negative expressions (thwarted expectations) within the reviews make it difficult for the machine learning techniques to accurately predict the emotions .

Another work on the YouTube comments was done by Smita Shree and Josh Brolin where the authors proposed an unsupervised lexicon-based approach to detect sentiment polarity of user comments in YouTube. They adopted a knowledge driven approach and ready a social media list of terms and phrases expressing 6 the user sentiment and opinion. But their results also showed that recall of negative sentiment is poorer compared to the positives, which can flow from to the wide linguistic variation utilized in expressing frustration and dissatisfaction.

Other works have performed sentiment analysis of social networks like Twitter to point out that there exists a relationship between the moods of individuals to the result of events within the social, political, cultural and economic spheres. Another research on the social media sentiment analysis is completed by A.Kowcika et al. In their paper they propose a system

which is in a position to gather useful information from the twitter website and efficiently perform sentiment analysis of tweets regarding the Smart phone war. The system uses efficient scoring system for predicting the user's age. The user's gender is predicted employing a well-trained Naïve Bayes Classifier. Sentiment Classifier Model labels the tweet with a sentiment. KrisztianBalog et al. proposed in his paper a way to gather useful information from the twitter website and efficiently perform sentiment analysis of tweets regarding the Smart phone war. The system uses efficient rating system for predicting the user's age. Twitter Sentiment 8 Analysis: the great the Bad and therefore the OMG!, paper by EfthymiosKouloumpis et al. deals with the utility of linguistic features for detecting the sentiment of Twitter messages. They evaluate the usefulness of existing lexical resources also as features that capture information about the informal and artistic language utilized in micro-blogging. Another sentiment analysis of web text was done using the blog posts by Gilad Mishne et al.

One of the foremost prominent works in website classification was done by Daniele Riboni in the paper "Feature Selection for website Classification"[44]. They conducted various experiments on a corpus of 8000 documents belonging to 10 Yahoo! categories using Kernel Perception and Naive Bayes classifiers. Their experiments show the usefulness of dimensionality reduction and of a replacement structured oriented weighing technique. They also introduce a replacement method for representing linked pages using local information that creates hypertext categorization feasible for real-time applications.

Other classification works are just like the one done by Eibe Frank et al.[46] In their paper they propose an appropriate correction by adjusting attribute priors. This correction are often implemented as another data normalization step, and that they show that it can significantly improve the world under the ROC curve. They also show that the modified version of MNB is extremely closely associated with the straightforward centroid-based classifier and compare the 2 methods empirically.

Another work on the sentiment analysis of social media is completed using multimodal approach, discussed within the paper by Diana Maynard et al.[47]. They examine a specific use case, which is to assist archivists select material for inclusion in an archive of social media for preserving community memories, moving towards structured preservation around semantic categories. The textual approach they take is rule-based and builds on variety of subcomponents, taking under consideration issues inherent in social media like noisy ungrammatical text, use of swear words, sarcasm etc[1]Athar, A. (2014). Sentiment analysis of scientific citations (No.UCAMCL-TR-856). University of Cambridge, Computer Laboratory. The author used NB and SVM classifier and compute the accuracies of the system using an F-score. Macro F-scores using uni-gram mentioned within the research work is 48 percent. [2] Pang, B., Lee, L., Vaithyanathan, S. (2002, July). Thumbs up? sentiment classification using machine learning techniques. Author used label data for the purpose of classification, they preferred the supervised learning approach. For the purpose of classification, the Na¨ıve Bayes classifier is used.

In this work, they need used a dataset of movie reviews. [3]Sentiment analysis and opinion mining (Liu, 2012):- Sentiment analysis and opinion mining is the field of study that analyses people's opinions, sentiments, evaluations, attitudes, and emotions from written language. It is one among the foremost active research areas on natural language processing and is also widely studied in data pre-processing, Web mining, and text mining. [4]Deep Learning for Hate Speech Detection in Tweets by Pinkesh Badjatiya (IIIT-H) , Shashank Gupta (IIIT-H), Manish Gupta (Microsoft), Vasudeva Varma (IIIT-H) (June 1st,

2017):- One of the most useful applications of sentiment classification models is that the detection of hate speech. Recently, there are numerous reports of the tough lives of content moderation staff. Our experiments on a benchmark dataset of 16K annotated tweets show that such deep learning methods outperform state-of-the-art char/word n-gram methods. [5]Mehmood, K., Essam, D., Shafi, K. (2018, July).Sentiment Analysis System for Roman Urdu. In Science and Information Conference (pp. 29-42) .Springer, Cham. They used their data set which is based on Urdu reviews related to movies, politics, mobile, dramas and miscellaneous domains extracted using scrapers as well as manual. The data-set was then classified using different supervised learning classifiers and compare their results with each other.

3. METHODOLOGY

The purpose of the methodology is defined in this section. Our methodology is depicted in Fig. 1. First of all, we used the annotated dataset. We used python based machine learning library named Scikit-Learn for implementing the system. Scikit-Learn is a well-known machine learning library tightly integrated with Python language and provides easy-to-interact interface.

First of all our system reads the data stored in the file having (Tab Separated Values) format. After reading, pre-processing phase is applied to clean and prepare the data for the use of machine learning algorithms. Directly text data cannot be given to machine learning algorithms, it should be converted into a suitable type.

Using Scikit-Learn module named "count-vectorizer ", the text data firstly convert into numeric format and prepare the matrix of tokens count.

Now the data is ready for machine learning algorithms. Then 60% of data is splitted randomly to train the classifier and 40% for testing the classifier' accuracy.

We perform our experiments in two phases, firstly we just apply N-grams (Length 1-3) features on data and compute accuracies using F-score and accuracy score. Secondly, in order to improve the accuracy scores, we apply other features like (stop words & punctuation removal, lemmatization, etc.) along with n-grams and then again compute the accuracies. The latter approach helps to reduce the noise and complexity of the data. Thirty iterations of each experiment were conducted to compute average results and a total of six experiments were performed. After computing the accuracies of each phase.

We then select the best feature which is giving the best result and which classifier is better in a specific scenario.

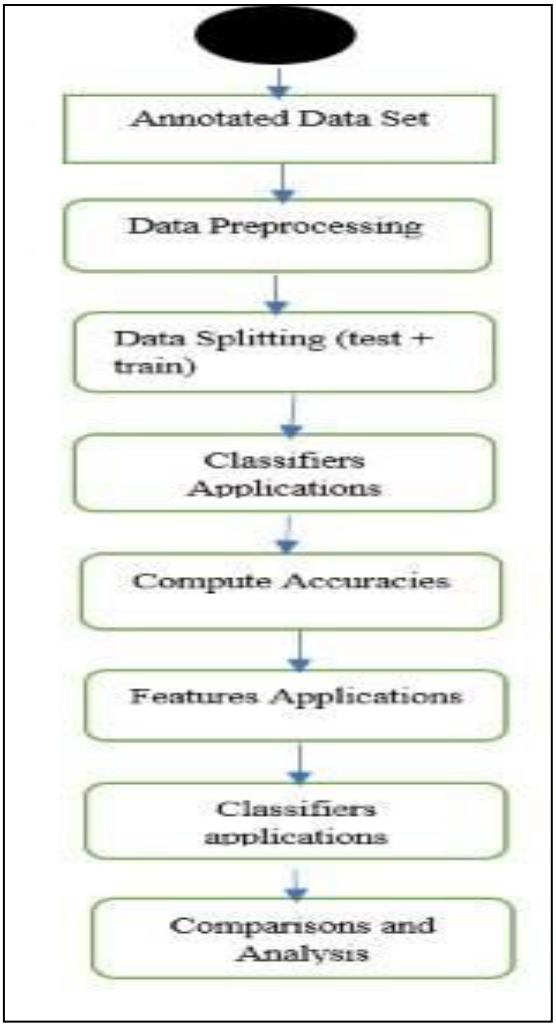

Fig. 1. Step by Step Flow of System Working.

3.1 EVALUATION METRICS

The evaluation of any research product decides the status and quality of that specific research work. This section briefly describes about the metrics used to evaluate the sentimental analysis system we developed. The performance of sentimental analysis system is evaluated by computing the accuracy of the classification results given by the system. Accuracy of the system is to be mentioned in the form of some units that include F-score and Accuracy score.

In our evaluation phase, we have calculated both Macro-F Score as well as Micro-F Score. Where FP is considered an error of type-1(false positive) and FN is considered an error of type-2(false negative). F-score is commonly used, a harmonic mean between precision and recall.

3.2 DATA PRE-PROCESSING

As corpus used for sentimental analysis classification is prepared or constructed. This data set is comprised of a total of 1500 citation sentences annotated as positive, negative, and neutral after applying rules. From total citation sentences, 60% of sentences were chosen randomly for training the classifier and the rest of 40% data was used for classifier' testing. The data set was cleaned to get the highest accuracy of the system.

A. Features Selection
For the sake of developing a system for sentiment analysis, different features are provided by ML .We have used various features e.g. lemmatization, n-grams, stop words and term-document frequency to evaluate the classifier' accuracy. Later the evaluation results will be displayed.

B. Lemmatization
Lemmatization is a process of normalizing the inflected forms of words. Homographic words cause ambiguity that disturbs searching accuracy and this ambiguity may also occur due to inflectional word forms . For instance, words like "Talking", "Talks" and "Talked" are the inflected forms of the word "Talk". The process of lemmatization and stemming is similar with minor changes, while the benefits of both approaches are the same. We have applied only lemmatization and avoid stemming due to the problems of stemming process. The stemming process is worthwhile for short retrieval lists, while our system has to deal with large data set and processing lists so we did apply stemming. Stemming performs normalization of inflected words by keeping different variations of words along with their derivation process

C. Stop Words and Punctuation
English text contains a lot of meaningless and non informative words called stop words. These are not required in classification because their presence just increase the size of data. So we applied stop words removal technique in order to cleanse the data for better and efficient classification . Some research works support the stop words removal from the data set to reduce the dimensions of data.

3.3 ALGORITHMS USED

This work attempted to utilize six machine learning techniques for the task of sentiment analysis. The modeling of all techniques is briefly discussed below.

➔ classification classifiers

After pre-processing and features selection the very next step is to apply classification algorithms. Many text classifiers have been purposed in literature. We have used 6 algorithms of machine learning including Naïve-Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbor (KNN), and Random Forest (RF).

a)Naïve Bayes: Naïve- Bayes is the most popular classification algorithm due to its simplicity and effectiveness. This classifier works according to the concept of Bayes theorem. It's a kind of module classifier that follows the idea of probabilities for the purpose of classification.

The benefit of using Naïve Bayes on text classification is that it needs less dataset for training. removal of numeric, foreign words, html tags and special symbols yielding the set of words. This pre-processing produces word-category pairs for training set. Consider a word 'y' from test set (unlabeled word set) and a window of n-words (x1, x2, …… xn) from a document.

The conditional probability of given data point 'y' to be in the category of n-words from training set is given by:

$$P(y/x1,x2,\ldots\ldots xn)=P(y)\times\prod i=1nP(xi/y)P(x1,x2,\ldots\ldots xn)$$

b) Support Vector Machine: In the world of machine learning one such supervised learning algorithm that achieves enough improvements on a variety of tasks is a Support vector machine classifier. Particularly in the case of analysing the sentiments.SVM algorithms had made excellent classifiers because the more complex the data will be the more accurate the prediction will be.

c) Decision Tree: In various fields of text classification the use of decision tree classifier can be seen and analysed. Its popularity is based on the nature of classification rules that make it interesting for NLP researchers. The decision is constructed by selecting the data from the data-set randomly.

Advantages are-Understandable prediction rules are created from the training dataBuilds the fastest treeBuilds a short treeOnly need enough attributes until all data is classified and the disadvantages are-

Data may be over-fitted or over-classified, if a small sample is tested, Only one attribute at a time is tested for making a decision, Does not handle numeric attributes and missing valuesTo prevent overfishing, we optimize the hyper-parameters of Decision Trees like max_features, min_samples_split,max_depth, etc.

d)Random Forest: mentioned the importance of a random forest classifier and compared its performance with the other classifiers claimed that the random forest algorithm provides efficient and discriminative classification, as a result, it is considered an interesting classifier.

e) K-th Nearest Neighbour: KNN is a simple and efficient classifier. Called lazy learner because its training phase contains nothing but storing all the training examples as classifiers. KNN requires a lot of memory while storing the training values.

the K-nearest neighbor algorithm essentially said that for a given value of K algorithm will find the K nearest neighbor of unseen data point and then it will assign the class to unseen data point by having the class which has the highest number of data points out of all classes of K neighbors.

3.4 FEATURES USED

Word clouds or tag clouds are the graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. The larger the word in the visual the more common the word was in thecomments. This type of visualization can assist evaluators with exploratory textual analysis by identifying words that frequently appear in a set of interviews, documents, or other text. It can also be used for communicating the most salient points or themes in the reporting stage.

3.5 SOURCE CODE AND IMPLEMENTATION

```
#importing liberaries

 import numpy as np
 import matplotlib.pyplot as plt
 import pandas as pd
 import seaborn as sns
```

```
#cleaning the dataset(removing stopwords,
stemming).

import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
  review = re.sub('[^a-zA-Z]', ' ',
dataset['Review'][i])
  review = review.lower()
  review = review.split()
ps = PorterStemmer()
all_stopwords = stopwords.words('english')
all_stopwords.remove('not')
  review = [ps.stem(word) for word in review if
not word in set(all_stopwords)]
  review = ' '.join(review)
corpus.append(review)
```

```
#Creating the Bag of Words model

from sklearn.feature_extraction.text import
CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, -1].values
```

```
#Splitting the dataset into the Training set and
Test set

from sklearn.model_selection import
train_test_split
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size = 0.20,
random_state = 0)
```

```
#Training the Naive Bayes, SVM , K-NN
model , Decision Tree model , Random Forest
model on the dataset

from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

from sklearn.svm import SVC
classifier = SVC(kernel = 'linear',
random_state = 0)
classifier.fit(X_train, y_train)

from sklearn.linear_model import
LogisticRegression
```

```
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)

from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)

#finally ,Making the Confusion Matrix of all the classificaion models

from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

here, will be creating ,wordcloudvisualizations of the comments in our dataset.

Basicallywordcloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using awordcloud.

 # for the visualization of positive sentiments on wordcloud we will store all the comments having polarity 1 in comments_positive.

Here the source code-

```
comments_positive=comments[comments['polarity']==1]
!pip install wordcloud
total_comments=''.
join(comments_positive['comment_text'])
wordcloud=WordCloud(width=1000,height=500,stopwords=stopwords).generate(total_comments)
plt.figure(figsize=(15,5))
plt.imshow(wordcloud)
plt.axis('off')
```

4.RESULTS

Different machine learning algorithms used for the classification. The evaluation metrics were used to validate the system. The detailed description of the experimental results using evaluation metrics is defined in Table I, and Table II. In these tables terms, A1, B1, C1 denotes simply unigram, bigram, trigram features while A2, B2, and C2 denote the application of unigram, bigram, trigram along with other features. Table I shows that Overall DT using n-grams gives the best F-score in macro while RF is best in case of micro average. LR is also overall best in the micro average without applying extra features. Uni-gram playssupport in better performance of LR and DT, uni-gram along with other features plays significant performance in NB, KNN, and RF. DT gives better performance in the case of uni-grams, bi-grams, and tri-grams. LR performance is significant in case of uni-grams only, k-th nearest neighbor outperforms in the case of n-grams along with the other features and give worst performs

without other features while RF performs best as same as KNN.

The overall discussion describes that uni-gram, bi-gram, and tri-gram without other features perform best where unigram is at first position. Table II shows that Overall SVM, LR, and RF performed very best with the highest accuracy scores. N-grams play significant performance in NB, SVM gives the best accuracy using uni-gram, LR performance is significant in case of bigrams and tri-grams, KNN outperforms in case of n-grams without other features and gives worst performs with other features. The overall discussion describes that uni-gram, bigrams, and tri-grams without other features performs best and give significant accuracy scores.

	NB		SVM		LR		DT		KNN		RF	
features	Macro scores %	Micro scores %	Macro scores %	Micro scores %	Macro scores %	Micro scores %	Macro scores %	Micro scores %	Macro scores %	Micro scores %	Macro scores %	Micro scores %
A1	36	87	37	88	49	87	49	85	33	87	44	88
A2	49	83	48	86	46	88	48	85	34	87	45	88
B1	34	87	31	87	46	88	49	86	32	87	44	88
B2	46	79	47	87	46	87	48	85	34	87	46	88
C1	36	87	31	87	44	88	49	86	32	87	42	88
C2	45	77	47	87	46	87	48	85	34	87	46	88

table I- f- scores

Features	NB%	SVM%	LR%	DT%	KNN%	RF%
A1	87	87	87	87	87	88
A2	84	88	84	87	86	88
B1	87	88	88	87	87	88
B2	79	87	85	87	86	88
C1	87	88	88	87	87	88
C2	77	88	86	87	87	88

table-II- Accuracy scores

```
In [21]: plt.figure(figsize=(15,5))
         plt.imshow(wordcloud)
         plt.axis('off')
Out[21]: (-0.5, 999.5, 499.5, -0.5)
```

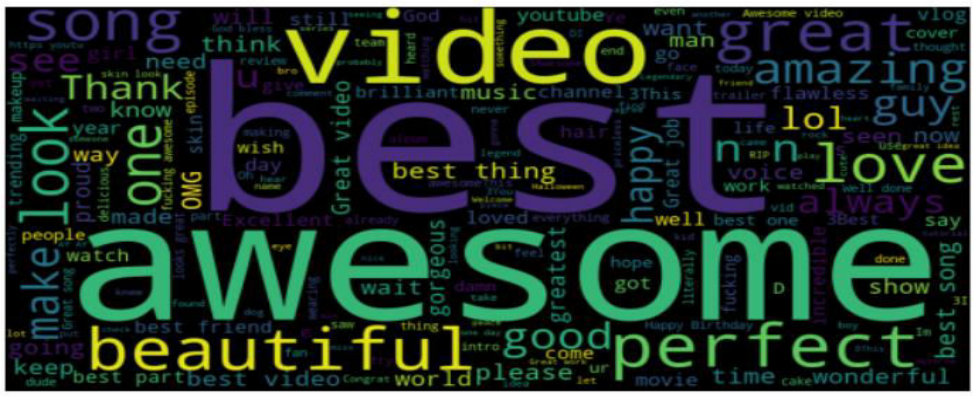

Figure-II Results of word cloud visualizations implement on our dataset

5.CONCLUSION

In this research work, we presented a sentiment analysis system for you-tube comments. We have used different machine learning classifiers which are NB, SVM, DT, LR, KNN and RF along with different features to process the data and optimize the classification results. Experiments are performed on the data set . Data set is partitioned into training and testing sets according to the ratio of 80:20 and. Accuracies of the classifiers are computed by using various evaluation metrics like F-score, and Accuracy score. The result shows that SVM performs better than other classifiers. After SVM Naïve Bayes performs well. In the case of the macro average, the performance of SVM classifier is best while computing Fscore, and accuracy measures while the random forest is best in case of micro average. Uni-grams, bi-grams, and tri-gram features performed very well and support the classifiers to achieve highest accuracy scores..We alsoused the n-grams approach together with the lemmatization process to reduce data dimensions . In this paper, we have implemented six classifiers LR, DT, KNN, and RF including NB, SVM . We computed the accuracies by increasing the number of evaluation metrics F-score and accuracy including F-score to evaluate the accuracies with the base system. Our results showed significant improvement like in the case of Naïve Bayes using uni-gram feature we achieved micro-F 87% while the base system described the result of micro-F = 78% and our results are approximately 9 % better. we achieved the macro-F = 49% by reducing the data dimensions by using the lemmatization process and stop words removal mechanism. Based on bi-gram and tri-gram features our system achieved the same result of micro-F = 87%. In our case, the micro-F based on bi-gram and tri-gram features increased by 11 %. while we improved our results to extant and achieved maximum of (micro-F = 87%, macro-F = 49%) that shows the significant improvement of our work.

6.REFERENCES

[1] Athar, A. (2014). Sentiment analysis of scientific citations (No. UCAMCL-TR-856). University of Cambridge, Computer Laboratory.

[2] Athar, A., Teufel, S. (2012, July). Detection of implicit citations for sentiment detection. In Proceedings of the Workshop on Detecting Structure in Scholarly Discourse (ppp. 18-26).

[3]Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... &Vanderplas J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research.

[4] Poria S, Cambria E, Gelbukh A, Bisio F, Hussain A (2015) Sentiment data flow analysis by means of dynamic linguistic patterns

[5] Turney PD, Mohammad SM (2014) Experiments with three approaches to recognizing lexical entailment.

[6] Parvathy G, Bindhu JS (2016) A probabilistic generative model for mining cybercriminal network from online social media: a review.

[7] Qazvinian, V., &Radev, D. R. (2010, July). Identifying non-explicit citing sentences for citation-based summarization. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 555-564). Association for Computational Linguistics.

[8]. Socher R (2016) deep learning for sentiment analysis—invited talk. In: Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis.

[9] Sobhani P, Mohammad S, Kiritchenko S (2016) Detecting stance in tweets and analyzing its interaction with sentiment. In: Proceedings of the 5th joint conference on lexical and computational semantics.

[10] Saif, H., He, Y., & Alani, H. (2012, November). Semantic sentiment analysis of twitter. In International semantic web conference (pp. 508-524). Springer, Berlin, Heidelberg.

[11] Dashtipour K, Poria S, Hussain A, Cambria E, Hawalah AY, Gelbukh A, Zhou Q (2016) Multilingual sentiment analysis: state of the art and independent comparison of techniques

12] EfthymiosKouloumpis, Theresa Wilson, and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of the Fifth International Conference on Weblogs and Social Media, ICWSM, (2011).

[13]. Cambria E, White B (2014) Jumping NLP curves: a review of natural language processing research.

[14]Mohammad SM, Zhu X, Kiritchenko S, Martin J (2015) Sentiment, emotion, purpose, and style in electoral tweets.