Opinion Extraction using Hybrid Learning Algorithm with Feature Set Optimization Approach

Abstract: - Evolution in engineering and technology added large size of data storage and transmission through the web application over the internet. This huge amount of data primarily used for exchange of information in between users and devices and in secondary aspects it has utilization as feedback, ratings and reviews that is supporting in generation of useful information of products, services, incidents etc. The data as opinion, feedback, view & suggestion is explored, organized & analyzed for selection of appropriate options. Sentiment analysis using the opinion extraction is a challenging task that is based on feature extraction and the concepts of Natural Language Processing that is applied in identification of the opinions of a user in terms of positive, neutral or negative ratings hidden in the form of comments typed as the text. Presently many data-processing based feature evaluation techniques for opinion extraction are used for solving the issues faced under sentiment classification applications. This article is based on development and application of algorithms for opinion extraction from text data available on web resources by K-Nearest Neighbor (KNN), Support vector machine (SVM) and hybrid of both named as SVM+KNN for classification of multi-label opinions from extracted text from review data of Twitter and Amazon. The performance of all the classification models (KNN, SVM and SVM+KNN) on both datasets is evaluated in terms of different parameters.

Keywords: Feature Extraction, KNN, SVM, Opinion Mining, Sentiment Classification, Text Mining.

I. INTRODUCTION

Social networking sites and online shopping platforms are performing as a huge and global platform for public to freely express the opinions, feelings, discussions on wide number of social, financial and technical issues. Users are free to provide reviews & feedbacks on products, services on online social networking platforms like instagram, twitter, facebook and e-commerce sites like amazon, flipkart etc. Ratings given by the customer in terms of stars or number of likes or shares along-with the comments or reviews on these web applications in form of text helps the commercial organizations and public to know about quality of goods and services. Presently digital marketing circulated by all the small or big organization and all of them are facing a transition from offline to online marketing domain; it has passed through a challenge of handling large amount of unstructured data. It is becoming an important target to transform unstructured data in helpful patterns that may be integrated to task of drawing useful decisions about product and services [1]. Commercial service circulates advertise or survey on social networking sites (Twitter, Instagram & Facebook) for collecting user opinions in efficient manner [2]. User opinion helps in estimating the level of satisfaction & tracking the competition. It is also useful to public in survey of goods and service available in market. Opinion extraction used to perceive the market trends in efficient manner that is used in updating the quality [3, 4]. The service providers, manufacturers, distributor and consumers views undergo through impact of response extracted from opinion of users. These opinions finally impact the stock market and economy of a country. Social networking platforms are growing rapidly & getting easier in terms of interaction of users. As a result, the opinion of public is spreading swiftly over the world and changing the views of followers significantly [1]. Hence decisions extracted from opinion as ratings or sentimental bringing transformation in associated planning of organizations [5].

Services provided by different sectors are reflected on comments publically available on Twitter, facebook etc. and works as significant information about service provided to citizens by the professional systems. Opinions of public shared as feeling or thoughts on several instances that has prevalent impact on social life [6]. For example, in health sector the patients sometimes face isolation from society and undergoes through seriously transformation in mental state that attains the stage of depression. In such cases experts may use automated extraction of sentiment using the text-based information shared by patient on web applications for systematic analysis [7]. Social networking sites openly share the opinions as the text data in terms of sentiments, emotions and feelings in the user’s posts and if
somebody has faced issues from a particular event, service or product may be tracked by estimating his level of satisfaction state.

Opinion extraction based on text data is useful because user comment plays also plays an important role in academic sector. The teacher performance may be estimated by in terms of enthusiasm, dedication & talent using the text-based comments provided by student. It is an effective approach for improving the teaching skills on online and offline class platform [8]. Unstructured text data-based feedback is difficult to observe and a lot of problems are faced in drawing useful conclusions manually. The automated approach for drawing sentiments from text data using opinion extraction by machine learning methods assist the domain covered under academic sectors and healthcare sectors for opting corrective steps. The social media platforms like Twitter & Facebook know incorporating the promotions activities associated to marketing and advertising for different organizations. Consumer and service provider both follow the considerable online analysis for knowing the potential of concerned service or product. Blogs and forums followed by users of similar domain of interests & help in assessment of the standard of various schemes circulated about products and services under different types of promotion strategy. In such application the opinion extraction-based analysis in terms of features extracted from unstructured text data helpful everyone to select the best option [9, 10]. Opinion extraction analysis is incorporating different methodologies and techniques on the basis of artificial intelligence and optimization schemes. All schemes have particular advantages as well as drawbacks. These feature-based techniques for opinion extraction using artificial intelligence and machine learning faces significant challenges that are including the ridicule, context, statements that are conveying emotions, spread of lexical, slang and syntax-based ambiguity [11]. Since no standard rules are set up in communication to express the opinion. Some methods focus on effect of post, some analyze sentiments, and some are performing on analysis over logical structure of message. Such variety of challenges in opinion extraction process under natural language processing requires an approach that gives performs efficiently. Remaining of the paper describe as follows: section II involves the related work, section III involves the method and material, section IV contains result and discussion and section V contains conclusion.

II. RELATED WORK

Worked on Facebook comments performed for multilingual texts using Machine Learning (ML) the F1-score parameter shows high value but the performance on non-English text was low. Author suggested for use of hybrid scheme to avoid this limitation [12]. An approach based on Lexicon was followed on text data in Chinese language-based opinion extraction on feedback over online website of poetry. Further improvement in accuracy is suggested using machine learning techniques [13]. A CNN and Bi-LSTM based ML approach proposed for achieving high F1-score using small dataset to focus on the problem related to feature extraction issues. The improvement may be further incorporated on analysis with different deep learning algorithm [14]. An emotion extraction work was performed for ISEAR dataset using hybrid learning model but the achieved accuracy was low due to poor classified of neutral emotion. Further improvements may expect on using different deep learning approaches [15]. A deep machine learning scheme proposed for opinion extraction for data collected from WhatsApp Chat. The performance in terms accuracy was medium due to presence of emojis in text [16]. Naive Bayes (NB) and KNN applied on extraction of sentiment on Twitter Dataset. The NB performed better than KNN but it had a drawback of improper working for unsupervised learning mechanisms [17]. A machine learning based approach followed on SVM on ISEAR dataset to improve performance in opinion extraction by natural language processing [18]. A machine learning algorithm developed for Emotion Line dataset the demonstrated result shows high F1-score value for extraction on two different datasets but data considered in this work was small [19].

III. METHODOLOGY

In this proposed work an implementation is performed for extracting information from text data to generate features in classification of two datasets belongs to Twitter and Amazon. The datasets consist of text-based comments as opinion of users. In the twitter dataset opinion expressed as four different types of sentiments and in the Amazon dataset the reviews are associated with star rating. Both datasets are large I size (8040 posts (text message)) of different users. This dataset is first of all passed through the preprocessing steps that include the tokenization of posts that breaks the sentence into individual words [27]. The second step includes removing the punctuation from the tokenized words from each post. In the third step "bag of words created". The Bag contains the vocabulary that has unique words and counts of the number of times a word is repeated in each post [28]. After creating the word, the next step under data preprocessing involves the process of removing the stop words. The term stop words
represent the words as token like 'to', 'and', 'of' etc. They do not reflect any kind of sentiments or emotions. After removal of stop words the words that are rarely used are also removed. This step is called as removal of infrequent words [29].

After the preprocessing of data finally a bag is created that consist of those words that frequently used in the data set. Thus, the Term frequency of each word is calculated (TF) and Inverse document frequency (IDF) is also calculated and saved in a sparse matrix. In this way TF-IDF feature array is generated.

A. Data Processing: After generation of TF-IDF feature the dataset is used for development of machine learning model using KNN. The dataset is split into training and testing sets. About 80% of data used for training purpose [25]. The training is performed by random selection of feature vectors at different combination of number of top words and word frequency. Several K-NN models are developed with different training sets and performance is evaluated in terms of model accuracy. The text data as comments are imported from CSV file to MATLAB software. The twitter data has labels as 4 types of sentiments [happiness, relief, enthusiasm, surprise] the data size: 8040 tweets. The AmazonCellPhone.csv file has class label as 5 different user rating [1, 2, 3, 4, 5] the data size is 20000 reviews. After importing the data, the step applied for tokenization of the text document. In this step the document represented as a collection of words (also known as tokens).

B. Remove punctuation from the tokenized documents. After removing punctuation next step of create Bag-of-words is followed. A bag-of-words is a data structure stores the calculated term-frequency. It records the number of times that words appear in each document of a collection.

Number of tweets = Number of documents = [NumDocuments]1x1=D
Number of unique words in all documents: [NumWords]1x1=W
Collection of all unique words = [Vocabulary]1xW
CountsxW = Number of counts a word (term) repeated in each document.

<table>
<thead>
<tr>
<th>Table 1: Count of word repeated in a document</th>
</tr>
</thead>
<tbody>
<tr>
<td>finally made it to phoenix! I am home.</td>
</tr>
<tr>
<td>8 words in</td>
</tr>
<tr>
<td>1. (1,1)</td>
</tr>
<tr>
<td>2. (1,2)</td>
</tr>
<tr>
<td>3. (1,3)</td>
</tr>
<tr>
<td>4. (1,4)</td>
</tr>
<tr>
<td>5. (1,5)</td>
</tr>
<tr>
<td>6. (1,6)</td>
</tr>
<tr>
<td>7. (1,7)</td>
</tr>
<tr>
<td>8. (1,8)</td>
</tr>
<tr>
<td>9. (1,9)</td>
</tr>
<tr>
<td>10. (1,10)</td>
</tr>
<tr>
<td>11. (1,11)</td>
</tr>
<tr>
<td>12. (1,12)</td>
</tr>
<tr>
<td>13. (1,13)</td>
</tr>
<tr>
<td>14. (1,14)</td>
</tr>
</tbody>
</table>

From the bag of words, the "stop words" are removed. Words like "a", "and", "to", and "the" (known as stop words) can add noise to data. This step is applied to remove stop words before analysis. After this the Infrequent Words are removed. Remove words with low counts from bag-of-words. Words with counts less than 1 are removed from total unique words. Here T is called as number of top words.

C. Hybrid Learning: The SVM+KNN as hybrid learner used in this paper applied Error correcting output codes (ECOC) for performing classification task. It is motivated by coding theory where transmitted information is encoded by binary strings. Presently this method upgraded for handling learning problem to increase ensemble diversity as hybrid learner. In this SVM_KNN classifier, each class assigned to codeword and a L learner trained in hybrid as binary classifiers constructed by columns of ECOC matrix such that rows represents class codewords and column represents partition of the dataset by classes merging of similar bit value. Decoding is referred as classification perform by matching codeword predicted by L with the class codeword nearest in Hamming distance. This SVM+KNN is generalized form of one-vs-one and one-vs-all classifier and as a hybrid algorithm, it effectively performs classifiers independent of errors in random sampling.
Algorithm

Set data distribution: Training\textpercent
Set number of top words: N_{top}
Set number for frequency of words: N_{frequency}

1: Data preparation

import data
read data file
all\texttweets <= extract text data
all\textsentiments <= extract sentiment labels
data_{1}<=break text sentence into token of words (tokenization)
data_{2}<=erase punctuation from data

2: Build a Bag of Words containing all tokenized tweets (ignore punctuation)

N_{W}<=count total number of words
N_{D}<=count total number of documents
V_{ANW}<=create vocabulary of unique words
Counts_{W\times D}<=assign address to each unique word in each document
bag<= form bag of Words of data; as [Number of words, number of documents, vocabulary, counts];
bag_{1}<=remove stop words from bag of words “bag”
bag_{2}<=remove infrequent words that are repeated less than N_{frequency} time.
top_{words}<=extract top words having repetition frequency > N_{top}

3: Features and labels

TF_{t,d} <= calculate number of occurrences of term t in document d
DF_{t} <= Calculate number of documents containing the term t.
W_{t,d} <= weight of term t in document d <= TF_{t,d} * log (N/DF_{t})
tf(t,d) <=calculate term frequency <= (TF(t,d)/\text{sum (TF(t,d))})
idf (t, d) <=log (ND/|\{d :d \in D \text{ and } t \in T\}|)
M_{1}<=tfidf(t,d,D)<= tf(t,d)*idf(t,d)

4: Data Distribution

define: m<= percent train <= [80\%, 70\% or 60\%] percent training data
n <= size of all tweets \% size of all data in the document
n-m <=size of testing Tweets
\%\% Create a feature matrix for training by selecting the m rows of the TF-IDF matrix and all columns
m_{i}=generate m random permutation of tweet address in between 1 to n integer value
train_{x}:=training_features<=randomly select m tfidf value as features from bag M_{1} w.r.t m_{i} address locations
train_{y}:=training_labels <=create a corresponding label vector with the first m from all\textsentiments\textvector
create one feature matrix for testing by selecting all rows of the TF-IDF matrix after row n (i.e. the remaining rows)
test_{x}: testing_features>= Bag M_{1}(from m to n data at remaining random address of m_{1})
\%\% from m to n data at remaining random address of m_{1} create a corresponding sentiment class label vector
test_{y}:testing_labels <= all\textsentiments_{yy}

5: K-Nearest Neighbor

\%\% call KNN function fitKNN and generate KNN based classification model.
knnmodel <= fitKNN(train_{x},train_{y})
predictions_{x} <= predict output labels using knnmodel for testing data test

6: Call SVM function fitSVM and generate SVM based classification model.
SVMmodel <= fit\text{\textunderscore SVM}(\text{train}_x, \text{train}_y)
predictionsB <= predict output labels using SVMmodel for testing data test_x

7: Call SVM\text{\textunderscore KNN} hybrid function fit\text{\textunderscore SVMKNN} and generate hybrid SVM\text{\textunderscore KNN} based classification model.

SVMmodel <= fit\text{\textunderscore SVMKNN}(\text{train}_x, \text{train}_y)
predictionsC <= predict output labels using SVMmodel for testing data test_x
label_1 <= find test label = 'relief'
label_2 <= find test label = 'surprise'
label_3 <= find test label = 'enthusiasm'
label_4 <= find test label = 'happiness'
Confusionchart <= generate confusion matrix from \{label_1, label_2, label_3, label_4\}

[TP, TN, FP, FN] <= calculate true positive, true negative, false positive, false negative value from confusion matrix
precision <= TP/(TP+FP)
recall <= TP/(TP+FN)
f1score <= (2*precision*recall)/(precision+recall)
accuracy <= (TP+TN)/(TP+TN+FP+FN)
Go to Step 1 and repeat step 1 to step 48

Steps involved in development of opinion extraction from text data:
Step 1: Set data distribution: Training\text{\textunderscore percent}
Step 2: Set number of top words :N\text{\textunderscore top}
Step 3: Set number for frequency of word :N\text{\textunderscore frequency}
Step 4: Data preparation
Step 5: Build a Bag of Words containing all tokenized tweets (ignore punctuation)
Step 6: Calculate Features and define sentiment labels
Step 7: Perform Data Distribution
Step 8: Create a feature matrix for training data
Step 9: Create a corresponding sentiment class label vector from remaining random address
Step 10: Call KNN function fit\text{\textunderscore KNN} and generate KNN based classification model.
Step 11: Call SVM function fit\text{\textunderscore SVM} and generate SVM based classification model.
Step 12: Call SVM\text{\textunderscore KNN} hybrid function to generate hybrid SVM\text{\textunderscore KNN} based classification model.
Step 13: Predict output labels using SVMmodel for testing data test_x
Step 14: Generate confusion matrix from \{label_1, label_2, label_3, label_4\}
Step 15: Calculate true positive, true negative, false positive, false negative value from confusion matrix
Step 16: Calculate precision, recall, F1score, accuracy
Figure 1: Flowchart of proposed work
IV. RESULTS

Before the simulation is performed on MATLAB software using Natural Language processing and Machine learning toolbox. The algorithm run for classification using KNN, SVM and SVM+KNN for different combinations of number of top words and word frequency.

(a) Average percent accuracy at 60% training length for twitter dataset.

(b) Maximum percent accuracy at 60% training length for twitter dataset.

(c) Average percent accuracy at 70% training length for twitter dataset.

(d) Maximum percent accuracy at 70% training length for twitter dataset.

(e) Average percent accuracy at 80% training length for twitter dataset.

(f) Maximum percent accuracy at 80% training length for twitter dataset.
**Figure 2** Results in terms of average percent accuracy using KNN, SVM and SVM+KNN for twitter dataset.

(a) Average percent accuracy at 60% training length for Amazon dataset.

(b) Maximum percent accuracy at 60% training length for Amazon dataset.

(c) Average percent accuracy at 70% training length for Amazon dataset.

(d) Maximum percent accuracy at 70% training length for Amazon dataset.

(e) Average percent accuracy at 80% training length for Amazon dataset.

(f) Maximum percent accuracy at 80% training length for Amazon dataset.

**Figure 3** Results in terms of average percent accuracy using KNN, SVM and SVM+KNN for Amazon dataset.
Table 2 Maximum and average percent accuracy for Twitter and Amazon dataset

<table>
<thead>
<tr>
<th>Training Length</th>
<th>Max</th>
<th>Average</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KNN</td>
<td>SVM</td>
<td>KNN&amp;SVM</td>
<td>KNN</td>
<td>SVM</td>
<td>KNN&amp;SVM</td>
<td>KNN</td>
</tr>
<tr>
<td>4824</td>
<td>0.7659</td>
<td>0.7649</td>
<td>0.7678 (130-140)</td>
<td>0.7638</td>
<td>0.7628 (120-130)</td>
<td>0.7668 (140-150)</td>
<td>0.7688</td>
</tr>
<tr>
<td>5628</td>
<td>0.7937</td>
<td>0.7917</td>
<td>0.7957 (140-150)</td>
<td>0.7926</td>
<td>0.7916 (140-150)</td>
<td>0.7946 (150-160)</td>
<td>0.7966</td>
</tr>
<tr>
<td>6432</td>
<td>0.7927</td>
<td>0.7907</td>
<td>0.7947 (140-150)</td>
<td>0.7916</td>
<td>0.7906 (140-150)</td>
<td>0.7936 (150-160)</td>
<td>0.7956</td>
</tr>
</tbody>
</table>

Table 3 Performance in terms of Precision, Recall and F1score for Twitter and Amazon dataset

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Top words &amp; word frequency</th>
<th>Training Length</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>140-70</td>
<td>60%</td>
<td>0.8015</td>
<td>0.8304</td>
<td>0.8159</td>
<td>0.8304</td>
</tr>
<tr>
<td>SVM</td>
<td>50-70</td>
<td>60%</td>
<td>0.8257</td>
<td>0.8546</td>
<td>0.8403</td>
<td>0.8546</td>
</tr>
<tr>
<td>SVM+KNN</td>
<td>120-50</td>
<td>60%</td>
<td>0.8257</td>
<td>0.8546</td>
<td>0.8403</td>
<td>0.8546</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Amazon data set</th>
<th>Training Length</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>140-180</td>
<td>70%</td>
<td>0.9303</td>
<td>0.9265</td>
<td>0.9284</td>
<td>0.9284</td>
</tr>
<tr>
<td>SVM</td>
<td>140-180</td>
<td>70%</td>
<td>0.9303</td>
<td>0.9265</td>
<td>0.9284</td>
<td>0.9284</td>
</tr>
<tr>
<td>SVM+KNN</td>
<td>140-180</td>
<td>70%</td>
<td>0.9303</td>
<td>0.9265</td>
<td>0.9284</td>
<td>0.9284</td>
</tr>
</tbody>
</table>

Table 4 Comparison of SVM, KNN, SVM+KNN Model with State-of-the-Art Methods

<table>
<thead>
<tr>
<th>Technique</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SABSA [26]</td>
<td>0.858</td>
<td>0.839</td>
<td>0.8485</td>
<td>0.837</td>
</tr>
<tr>
<td>SentVec [21]</td>
<td>0.877</td>
<td>0.858</td>
<td>0.8675</td>
<td>0.861</td>
</tr>
<tr>
<td>Ngram+TF-IDF+ SVM [23]</td>
<td>0.866</td>
<td>0.846</td>
<td>0.856</td>
<td>0.844</td>
</tr>
<tr>
<td>SEML [25]</td>
<td>0.854</td>
<td>0.837</td>
<td>0.8455</td>
<td>0.838</td>
</tr>
<tr>
<td>MTMVN [24]</td>
<td>0.817</td>
<td>0.789</td>
<td>0.803</td>
<td>0.792</td>
</tr>
<tr>
<td>SVM+KNN (Twitter Dataset)</td>
<td>0.846</td>
<td>0.903</td>
<td>0.873</td>
<td>0.869</td>
</tr>
<tr>
<td>SVM+KNN (Amazon Dataset)</td>
<td>0.946</td>
<td>0.918</td>
<td>0.932</td>
<td>0.854</td>
</tr>
</tbody>
</table>

V. Conclusions

The SVM+KNN based hybrid scheme gave highest accuracy with respect to applying only SVM or KNN only. High accuracy for classification based on opinion extraction from posts of twitter or reviews of user on Amazon are used to for generating TF-IDF based feature. The highest accuracy is 87 % for twitter dataset and 85% on Amazon data set is observed for task of deciding opinion in terms of sentiments/rating by using textual content. The analysis is focusing the ambiguity covered in observing sentiments/rating is the main concerning challenge that is reduced by using hybrid learning scheme. The results are reflecting potential of natural language processing for opinion extraction in the post (Tweets/reviews) as text data. The approach may help to provide valuable applications on dataset processing prior to classification. In future advanced refinement steps for dataset may be used for further improving the performance. Detection of inaccurate or mislabeled classes may be focused for further advancements for understanding linguistic and sentiment analysis.
ACKNOWLEDGMENT

We would like to thank the university's research departments for providing the Manuscript number IU/R&D/2024-MCN0002660 according to their guidelines. This identification makes it easier to track and communicate about our research as it moves through the publication process. We would also like to express our appreciation to everyone who helped to build this work.

REFERENCES


