Abstract—Sentiment analysis over Twitter offer organisations a fast and effective way to monitor the public feelings towards their brand, business etc. A wide range of features and methods for training sentiment classifiers for Twitter datasets have been researched in recent years with varying results. In this paper, we experimented with various entities from tweets using word ngram model. We measured the correlation of the representative concept with negative and positive sentiment. We apply this approach to predict sentiment for four different Twitter datasets such as iphone, android, obama and amazon. Ngram model is evaluated with precision, recall and F measures. The results shows that the unigram model for identifying both negative and positive sentiment is a promising approach compared with other ngram models. We have evaluated the ngram model using different classification models such as Support Vector Machine (SVM), naive machine learning approach (NB), K-nearest neighbour approach (KNN) and Desition tree approach (DT). It has been considered as a classification problem and it has been modeled by means of vector representation.

I. INTRODUCTION

Nowadays social media, such as Twitter, produce a vast amount of information that lead us to new challenges in Machine Learning (ML) and in Natural Language Processing (NLP) fields. Twitter is a microblogging service, which according to latest statistics, has 284 million active users, 77% outside the US that generate 500 million tweets a day in 35 different languages. That means 5,700 tweets per second and they had peaks of activity of 43,000 per second. This numbers justify the great interest in the automatic processing of this information. The study estimates that 50.9% of tweets have some useful information that are capable of mobilize opinions in Internet and also in the real world. Therefore, social media users opinions have great strategic value for different organizations. Our work is focused on automatically identify the prevailing sentiment in a tweet using ML and NLP techniques. We developed a system for determining the tweets polarity to classify tweets among positive and negative. We represented features extracted using a bag of n-grams. We used various machine learning models such as Support Vector Machine (SVM), naive machine learning approach (NB), K-nearest neighbour approach (KNN) and Desition tree approach (DT). It has been considered as a classification problem and it has been modeled by means of vector representation.

The remainder of this paper is organized as follows: prior works on sentiment analysis are discussed in Section 2. The proposed approach is detailed in Section 3. Then, experiments and results are given in Section 4 and conclusions and future scope is presented in section 5.

II. LITERATURE SURVEY

Sentiment Analysis has been widely studied in the last decade in multiple domains. Most work focuses on classifying the polarity of the texts as positive, negative, mixed, or neutral. The pioneering works in this field used supervised [Pang et al., 2002] or unsupervised (knowledge-based) [Turney, 2002] approaches. In [Pang et al., 2002], the performance of different classifiers on movie reviews was evaluated. In (Turney, 2002], some patterns containing POS information were used to identify subjective sentences in reviews to then estimate their semantic orientation. In [Pang and Lee, 2008] we can find a comprehensive study of the different techniques used to identify the polarity of a text. Many efforts have been made to transfer this knowledge to language extracted from social media. In the literature we can find recent attempts to solve this problem using different machine learning approaches such as, SVM, Maximum Entropy, Naïve Bayes, etc, [Barbosa and Feng, 2010, O’Connor et al., 2010a, Zhu et al., 2014]. At best, these works achieve F1-score close to 70%.
T. Jhansi Rani, International Journal of Technology and Engineering Science \( ^{\text{TM}} \) IJTES

Volume 4[1], pp: 7032-7035, 2016

III. PROPOSED APPROACH

In this model various steps are data preprocessing, feature extraction, classification based on sentiments and labelling a tweet with a sentiment either positive or negative. The data set is separated into training and testing set. In the first phase, various n gram features such as unigrams, bigrams, trigrams and tetragrams are extracted from the data. The test instances are created, on the basis of these features. In the second phase, an classification model is built from training data, so as to be tested on unknown test data. The training are numerical feature vectors which represent term frequency of every selected feature, followed by the sentiment label attached with it and test instances are numerical feature vectors which represent term frequency of every selected feature. Labeled training data are used to train a machine learner, as it allows the evaluation of classification.

Data preprocessing is a very important step in sentiment classification. Text documents in their original form are not suitable for meaning patterns generation. They must be converted into a suitable input format. It can be converted into a vector space since most of the learning algorithms use the attribute, value representation. This step is important for the next stages. Data preprocessing involves tokenization, stopword removal and stemming. Tokenization is the process of chopping a document into small units called tokens which usually results in a set of atomic ngrams. This phase outputs the article as a set of terms by removing the unnecessary symbols like semicolons, colons, exclamation marks, hyphens, bullets, parenthesis, numbers etc. A stop list is a list of commonly repeated features which appear in every text document. The common features such as pronouns, conjunctions and prepositions need to be removed because they do not have effect on the classification process. For the same reason, if the feature is a special character or a number then that feature should be removed. In this paper, empirical evaluations are carried using various lexical features like word ngram features because they are more reliable than semantic features. The different word level features considered in this paper are character unigram, bigram, trigram and tetragrams. Word unigram takes individual words as tokens where as word bigram considers two consecutive words, trigram consider three consecutive and tetragram considers four consecutive words with overlapping as features.

IV. RESULTS AND DISCUSSIONS

IV.i. DATA DESCRIPTION

Tweets were collected on four topics such as amazon, android, iphone6 and obama. These tweets were collected by using Twitter APIs. The REST APIs are used to access the data from Twitter and post data on the Twitter. It is also useful to read author profile and the details of the followers of the author.

New responses to REST API queries can be delivered from streaming APIs by establishing HTTP connection. Based on the search query matching, updates are received in sync with user profile updates. REST APIs the Streaming APIs is a better solution if the application is rate-limited for over-polling. Twitter offers several streaming endpoints such as public streams, user streams and site steams and each is customized to certain use cases.

The collected tweets were parsed using Natural Language Tool kit for stemming, tagging, syntactic parsing and term extraction. The terms are annotated with WordNet for grouping the related terms together.

IV.ii EVALUATION MEASURES

In order to compare the results of all possible features with classifiers, we computed the precision, recall and F1 measure. Precision is the proportion of examples labeled positive by the system that were truly positive, and recall is the proportion of truly positive examples that were labeled positive by the system. where F1 is computed based on the following equation:

\[
F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}
\]
where,

\[
\text{Precision} = \frac{X}{X+Y}
\]

\[
\text{Recall} = \frac{X}{X+Z}
\]

Where X is documents assigned and correct, Y is documents assigned but not current and Z is documents not assigned but correct. The precision, recall and F1 values obtained for different classifiers in combination with different word level features are presented in the below tables.

Precision, Recall and F1 values are calculated for various word level features on different classifiers. From the values obtained it can be concluded that word unigram feature is performing well out of all other word level features. From the view of classifiers, SVM performance is good compared with all other classifiers.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Positive Sentiment</th>
<th>Negative Sentiment</th>
<th>Average Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigram</td>
<td>61.03</td>
<td>71.73</td>
<td>66.38</td>
</tr>
<tr>
<td>Bigram</td>
<td>60.28</td>
<td>70.58</td>
<td>65.43</td>
</tr>
<tr>
<td>Trigram</td>
<td>59.58</td>
<td>69.75</td>
<td>64.67</td>
</tr>
<tr>
<td>Tetragram</td>
<td>59.72</td>
<td>69.81</td>
<td>64.76</td>
</tr>
</tbody>
</table>

I. Table 1: Averages of F measures across all four datasets using Naive Bayes

<table>
<thead>
<tr>
<th>Feature</th>
<th>Positive Sentiment</th>
<th>Negative Sentiment</th>
<th>Average Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigram</td>
<td>58.43</td>
<td>68.36</td>
<td>63.40</td>
</tr>
<tr>
<td>Bigram</td>
<td>57.12</td>
<td>67.82</td>
<td>62.47</td>
</tr>
<tr>
<td>Trigram</td>
<td>57.01</td>
<td>67.21</td>
<td>62.11</td>
</tr>
<tr>
<td>Tetragram</td>
<td>58.56</td>
<td>66.10</td>
<td>62.33</td>
</tr>
</tbody>
</table>

Table 2: Averages F measures across all four datasets using K nearest Neighbor

<table>
<thead>
<tr>
<th>Feature</th>
<th>Positive Sentiment</th>
<th>Negative Sentiment</th>
<th>Average Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigram</td>
<td>65.66</td>
<td>74.67</td>
<td>70.17</td>
</tr>
<tr>
<td>Bigram</td>
<td>63.82</td>
<td>73.55</td>
<td>68.69</td>
</tr>
<tr>
<td>Trigram</td>
<td>62.23</td>
<td>72.92</td>
<td>67.57</td>
</tr>
<tr>
<td>Tetragram</td>
<td>62.11</td>
<td>73.23</td>
<td>67.67</td>
</tr>
</tbody>
</table>

Table 3: Averages F measures across all four datasets using Support Vector Machine

<table>
<thead>
<tr>
<th>Feature</th>
<th>Positive Sentiment</th>
<th>Negative Sentiment</th>
<th>Average Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigram</td>
<td>55.68</td>
<td>67.37</td>
<td>61.52</td>
</tr>
<tr>
<td>Bigram</td>
<td>54.59</td>
<td>66.89</td>
<td>60.74</td>
</tr>
<tr>
<td>Trigram</td>
<td>53.25</td>
<td>66.21</td>
<td>59.73</td>
</tr>
<tr>
<td>Tetragram</td>
<td>53.12</td>
<td>64.96</td>
<td>59.04</td>
</tr>
</tbody>
</table>

Table 4: Averages F measures across all four datasets using Decision Tree
V. CONCLUSIONS

In this work, experiments tried to find the best ngram model for sentiment analysis using word level features features. Empirical evaluations are carried out on the test set using different machine learning classifiers such as naive bayes, k-nearest neighbour, support vector machine and decision tree classifiers in combination with different word ngram level features. From the results it can be concluded that the word unigram model are better than the other word based features. The word unigram feature gave the best F1 score obtained as 0.83 for sentiment classification. The SVM classifier shows good performance in this experiment of AA compared with all other classifiers. As a part of future work we may experiment with other machine learning algorithms with other types of features such as syntactic and semantic and also with the combination of different types of features.

REFERENCES