TWITTER SENTIMENTAL ANALYSIS

B.Tech Minor Project Report

BY

PAPANI SAI CHARAN (15115060)
VADDINENI PAVAN KUMAR (15115083)
THADISETTY SRINIJA (15115080)

SUPERVISED BY

Dr. SARSIJ TRIPATHI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
RAIPUR, C.G. (INDIA)
DECEMBER, 2018

TWITTER SENTIMENT ANALYSIS

A Minor Project Report

submitted in partial fulfilment of the
requirements for the award of the degree
of
Bachelor of Technology
in
COMPUTER SCIENCE & ENGINEERING

BY
PAPANI SAICHARAN (15115060)
VADDINENI PAVAN KUMAR (15115083)
THADISETTY SRINIJA (15115080)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
RAIPUR, CG (INDIA)
DECEMBER, 2018

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Minor Project Report entitled **"Twitter Sentimental Analysis",** in partial fulfilment of the requirements for the award of the Bachelor of Technology in Computer Science & Engineering and submitted to the Department of Computer Science & Engineering of National Institute of Technology Raipur is an authentic record of my own work carried out during a period from July 2018 to December 2018 under the supervision of **Dr. Sarsij Tripathi**, Assistant Professor, Department of Computer Science & Engineering, NIT Raipur.

The matter presented in this report has not been submitted by me for the award of any other degree elsewhere.

Signature of Candidate

VADDINENI PAVAN KUMAR
15115083

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: Signature of Supervisor

Dr. Sarsij Tripathi
Assistant Professor, CSE, NIT Raipur

Head of Department
Computer Science & Engineering
National Institute of Technology, Raipur

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Minor Project Report entitled **“Twitter Sentimental Analysis”,** in partial fulfilment of the requirements for the award of the Bachelor of Technology in Computer Science & Engineering and submitted to the Department of Computer Science & Engineering of National Institute of Technology Raipur is an authentic record of my own work carried out during a period from July 2018 to December 2018 under the supervision of **Dr. Sarsij Tripathi**, Assistant Professor, Department of Computer Science & Engineering, NIT Raipur.

The matter presented in this report has not been submitted by me for the award of any other degree elsewhere.

Signature of Candidate

PAPANI SAICHARAN
15115060

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: Signature of Supervisor

Dr. Sarsij Tripathi
Assistant Professor, CSE, NIT Raipur

Head of Department
Computer Science & Engineering
National Institute of Technology, Raipur

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Minor Project Report entitled **“Twitter Sentimental Analysis”,** in partial fulfilment of the requirements for the award of the Bachelor of Technology in Computer Science & Engineering and submitted to the Department of Computer Science & Engineering of National Institute of Technology Raipur is an authentic record of my own work carried out during a period from July 2018 to December 2018 under the supervision of **Dr. Sarsij Tripathi**, Assistant Professor, Department of Computer Science & Engineering, NIT Raipur.

The matter presented in this report has not been submitted by me for the award of any other degree elsewhere.

Signature of Candidate

THADISETTY SRINIJA
15115080

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: Signature of Supervisor

Dr. Sarsij Tripathi
Assistant Professor, CSE, NIT Raipur

Head of Department

Computer Science & Engineering
National Institute of Technology, Raipur

ACKNOWLEDGMENT

We would like to thank our project advisor, Dr. Sarsij Tripathi, Assistant Professor, Dept. of CSE, NIT Raipur, for his inspiring guidance, constructive criticism, valuable suggestions, and support. We also thank him for constant supervision as well as for providing necessary information regarding the project. We feel extremely grateful for the opportunity to spend a semester working on a project in a field of great personal interest under his direction.

Every project is having it's own importance in it's respective field. Every attempt is equally importance to form a chunk. We sincerely attempted to create this project by gaining the knowledge from internet and our professor. We are thanking our professor for our proper guidance.

Papani Saicharan (15115060)

Vaddineni Pavan Kumar (15115083)

Thadisetty Srinija (15115080)

ABSTRACT

Twitter Sentimental Analysis is one of the most informative in capturing the opinions and sentiments of people to tune the product quality. We have used twitter data as our dataset because it is not specific to one domain so that we can develop a model which predict the sentiment of tweet. Here we trained and tested the machine with Stanford dataset by improving pre-processing, improving feature extraction quality (Unigrams, bigrams, trigrams). Project includes the classifiers such as Naïve Bayes, Maximum Entropy, Decision Trees to improve accuracy of the sentiment in tweet. We will use negation words and negation scope of words in tweet to get better accuracy than existing models for sentimental analysis. Naïve Bayes classifier give the best accuracy in sentiment analysis of tweet, so along with Naïve Bayes we also used other classifiers like maximum entropy, decision trees to check whether we get best accuracy or not.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

1.1 TWITTER

Microblogging is the one of the most used tool by many people to express their view and opinions on current issues or products in short. Twitter is the best and easily adaptable microblogging website from the available one's (it is the most popular website too).Twitter allow the users to tweet the opinions with maximum of 140 characters, user is given freedom to use any type of slang, abbreviations, emotion icons and short forms. Even the popular news channel CNN sometimes displays the tweets count on it's channel while taking the interviews of political leader to give overall public view. We took the twitter to grab the dataset which we will be using for performing sentimental analysis.

In this paper we will be performing sentimental analysis for the tweets to get the sentiment of the tweet. This sentiment is used by many companies for advertising, improving and rating of products. It is also used to give the public opinion on the political issue. It will also be used for recommendation system.

1.2 SENTIMENTAL ANALYSIS

To say in simple words "it is the analysis of text for getting the sentiment". Sentiment may be either positive, negative, neutral (polarity). Beyond polarity classification is also possible like sad, happy and other.

Subjective and objective identification is also possible using sentimental analysis. This sometimes becomes more complicated than simple polarity classification. Subjectivity of words depends on the context in which they are used and Objective file may contain the subjective sentences and words but some research papers shown that by removing the objective sentences from file before classifying improved it's performance Pang [1].Existing approaches to sentimental analysis are:

- Knowledge based Technique
- Statistical Technique
- Hybrid Technique

Knowledge based technique classify based on unambiguous affect words like happy, good, bad, sad etc. Statistical methods based on machine learning approaches like SVM, Naïve approaches, bagging of words etc. Grammatical dependency relations, contextual dependences are also used i1.3n this approaches. Hybrid Technique is the mixture of both the above approaches. Here (elements of knowledge representation in hybrid) ontologies and semantic networks are used to get sentiment from tweets which is not clear. In this paper we used to statistical Techniques (SKLEARN library and NLTK [2]) as we need to process larges datasets.

Apart from above approaches, Sentimental analysis approaches broadly classified into two:

- Lexicon based
- Machine learning based

Lexicon based is an unsupervised approach which uses the lexicons for analysis and scoring methods for evaluating opinions. In machine learning approaches we choose an interesting dataset, perform pre-processing, feature extraction, statistical and mathematical models (machine learning models) etc. We can have either two level or three level classification tweet classification (pos, neg, neu).

1.3 WORKFLOW

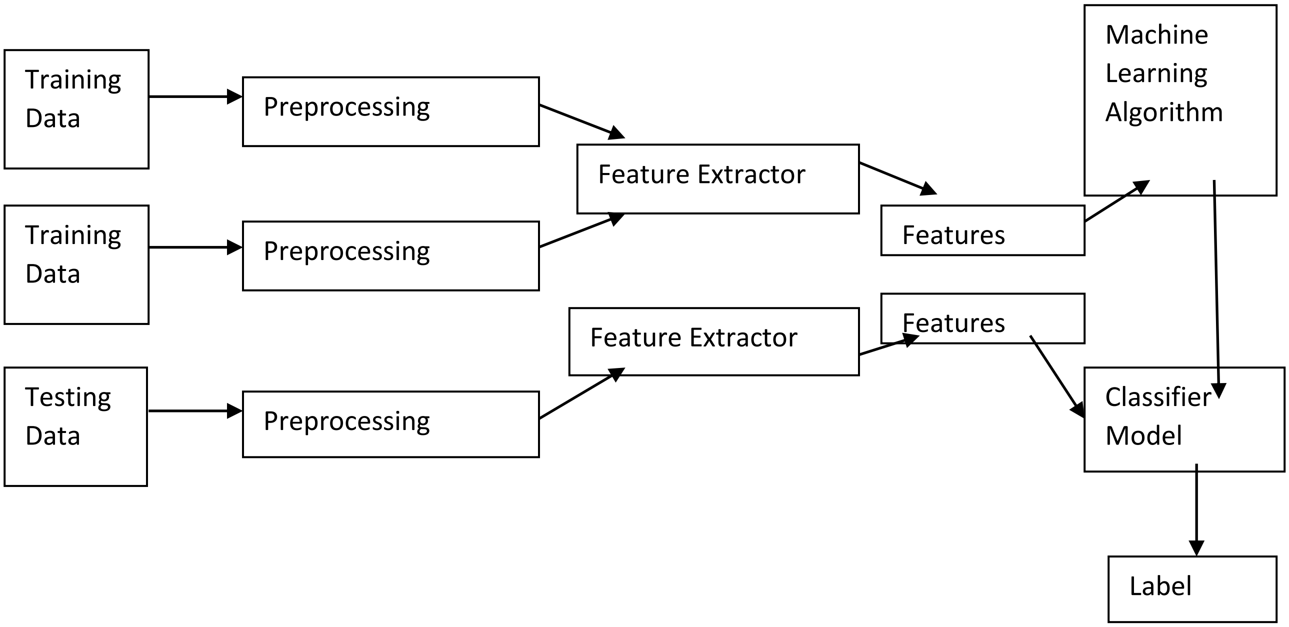


Figure 1. 1 : Workflow of Entire Sentiment Analysis

CHAPTER 2 - LITERATURE REVIEW

2.1 INTRODUCTION

Pak Paroubek [3] collected corpus of tweets consisting of +ve sentiment, -ve sentiment and objective texts (no setiments) using twitter API (from accounts of 44 newspapers like New York Times). It states that the frequency distribution of words followed ZIPF law which indicates good collection of corpus. Subjective vs and Objective plots with POS tags, Experimented with unigrams, bigrams, trigrams but Pang et al [4] stated that unigrams performed better than bigrams, Dave et al [5] got good results with bigrams and trigrams than unigrams for product-review polarity classification. Pak used multinomial Naïve Bayes classifier and tried SVM, CRF but at last Naïve classifier give better results. To increase accuracy Pak used two strategies

- Entropy
- salience

Higher the value of entropy indicates uniform distribution of n-grams in different sentiments. We use the lower value of entropy to increase the accuracy whereas salience value lies takes either 0 or 1.if salience value is 0 then n-gram is discriminated else considered. It also used threshold values in both strategies to improve accuracy. These two strategies are used to remove the n-grams that does not indicate any sentiment.

Another popular sentimental classification is done by Barbosa and Feng [6].They created 1000 label tweets manually for tuning and other 1000 which is also manually created for testing. They used prior polarity and POS of words in conjunction with punctuations, hashtags, retweets and other.

CHAPTER 3 - METHODOLOGY

3.1 DATASET

Before going deep, we shall discuss few words related to twitter:

Emoticons: pictorial representation of facial expression.

Hashtags: It is used to mark a topic.

Usernames: twitter user names are denoted with @ at the start of the names.

We used two datasets and performed the sentimental analysis

- Sanders Analytics LLC [7]
- Stanford dataset

Sanders Analytics [7] contain 5513 tweets collected from 4 different companies accounts (Apple, Microsoft, Google and other). It has to be download from online using Twitter API in the format of JSON and parsed accordingly.

Stanford dataset which consist of 1600000 data points, It consist of 5 columns like polarity, tweet-Id, Date and time, subject, username, text.

In this paper we used Stanford dataset and normalized accordingly, we build separate functions (python code) for casting the dataset for pre-processing.

3.2 REQUIREMENTS

Python [8]**:** it is a high level language and interpreted programming language. It is easily understandable and easy inundation to create blocks. Python provide large set of libraries which shows it is importance in various application like machine learning, data analysis etc.

NUMPY [9]**:** It is the famous package available in python for scientific computation with high performance array objects and arrays.

Natural Language Processing: It is a library in python is NLTK [2] which is used for text processing and classification. It also contain many trainable classifiers and can perform operation like tokenizing, POS tagging, creating bag of words.

3.3 PRE-PROCESSING

Pre-processing is one of the important step that completely alter the rate of accuracy in most cases. Here we normalize the text of tweets by series of pre-processing techniques. For applying the learning algorithm we require the dataset to be in adequate size so we reduce the size of the dataset accordingly, We also show statistics graphically about the datasets.

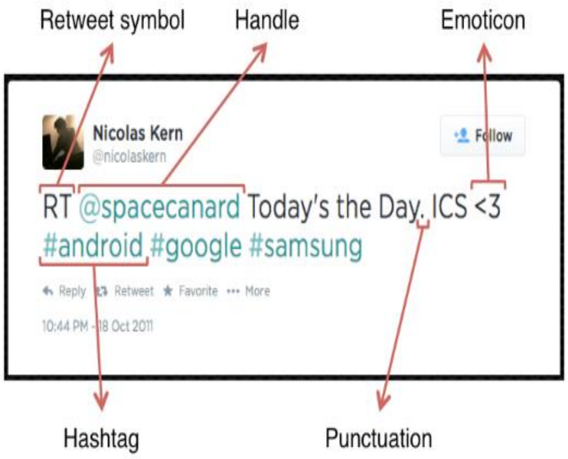
We use the pattern matching of python (regular expression module) to replace the below.

3.3.1 HASHTAGS

These are denoted using # in front of the topic and used to indicate a particular topic or current trend issue, we replace it with HASH_\1 for easy recognition in future.

3.3.2 HANDLES

Handles are denoted as ‘@username’. These are used to direct the written part to some user. It is replaced with HNDL_\1



RT @spacecanard Today's the Day, ICS <3 #android #google #samsung

FIgure 3.1 Twiiter Tweet Description

3.3.3 URLS

URL do not contribute much to the sentimental analysis of tweets in case of text classification. In some scenarios URL may be an important feature. We replace the URLS (http;//followmei.com) with the URL using regular expressions.

3.3.4 EMOTICONS

Emoticons	Examples			
EMOT_SMILEY	: -)	:)	(:	(-:
EMOT_LAUGH	:-D	:D	X-D	XD
EMOT_LOVE	<3	:*		
EMOT_WINK	;-)	;)	;-D	;D
EMOT_FROWN	:-(	:(	(:	(-:
EMOT_CRY	:,(	:'(	:"(	:((

Table 3.1 : List of Emoticons that can identified

Here we replace the emoticons with single word. Above listed emoticons are only identified not all.

3.3.5 PUNCTUATIONS

Punctuations gives us some important information regarding the sentiment. Not all will definitely contribute for but some definitely does (like question mark, exclamation mark etc.). Below listed punctuations are currently detected and replace with text.

Punctuations	Example
PUNC_DOT	.
PUNC_EXCL	!
PUNC_QUES	?
PUNC_ELLP	…

Table 3.2 : Punctuations that are identified in our feature Extraction

3.3.6 REPEATING CHARACTERS

Human are generally crazy, so they tune the slang of the language in most funny way like happyyyyyy, hungryyyyyyyyyyyy. So we replace such repeating characters with maximum of two characters(we use regular expression for pattern matching).

3.4 STEMMING ALGORITHMS

Stemming is kind of normalizing procedure. Disregarding the tense of words, similar meaning is associated with different variations of that particular word. “Fasten the lookup process and to normalize sentence” is the main reason to stem the words/sentence. Removal of affix, mixed and stats based are 3 most common types of stemming algorithms. The first method removal of affix is the fundamental one. This method removes most common suffices and/or prefixes by applying group of transformation rules. Less used stemming method is to cutoff words at N-th symbol(Which is not appropriate in practicality).

Example:

I was taking bath in bathtub.

I was bathing in bathtub.

Above two sentences mean same. Common part is “in bathtub” and “I was”. Only difference is “taking bath” was changed as “bathing” which do have same meaning. Differentiating between bathing and bath to find out what does this past tense activity is truly necessary? No, not really.

Imagine every affix and different tenses you can use to word in English language. Highly redundant and inefficient method is to have per each version of word having respective dictionary. Because when we use numerical representation for this words, all words do possess same value.

In 1968 J.B Lovins proposed first ever stemming algorithm.It describes 294 endings,each end being connected to one among 29 possible conditions ,added to that 35 transformation rules.End of word/sentence with matching condition is searched and truncated.

3.4.1 PORTER STEMMER

Porter Stemmer is the most popular stemming algorithm which had been proposed around 1979 by Martin Porter and published in 1980 July. It is most standard algorithm then and now used for stemming in English language. We get a good trade-off between correctness, readability, fastness. In 6 consecutive steps sixty rules are applied without any recursion.

Algorithm steps are:

- Removes –ed or –ing suffixes and plurals(s as a suffix).
- If there is 2 or more ovels in stem this algorithm turns terminal from y to i.
- Double suffixes are mapped to single ones: -ational,-ation and so on…
- Suffixes such as –ness, -ful, -full etc.. are properly dealt.
- Cuts off -ence, -ant,-ism and so on…
- Final –e is removed.

How to use in program:-

Firstly, Stemmer should be grabbed and defined:

```
from nltk.stem import PorterStemmer
From nltk tokenize import sent_tokenize
From nltk.tokenize import word_tokenize
P = PorterStemmer()
```

For example choose some words with similar stem, like:

Word_ex = [“program”,”programmer”,”programming”,”programmed”,”program-mmly”]

Stem those above words by using code in this way:

```
for s in words_ex :
 print(p.stem(s)).
```

Output:

```
program
program
program
program
programli.
```

Now let us consider stemming a sentence:

text = "Be careful while pythoning with python in pythonly way.”

wor = word_tokenize(text)

```
for words in wor :
        print(p.stem(words))
```

Result is:

```
Be
careful
while
python
with
python
in
pythonli
wai
.
```

.

3.4.2 LEMMATIZATION

In lemmatization our main focus is on word normalization than on just stem finding. Here in addition to suffix removal, suffix may also be replaced with different one. This process may also include tagging POS (parts of speech) for each word and then normalization rules are applied .Dictionary search may also be needed. Let us consider an example , verb ‘see’ is lemmatized format of ‘saw’ and noun ‘saw’ will not be changed to any other word after lemmatization. For our text classification, stemming is necessary and sufficient.

3.5 FEATURES

Classifier for tweets can be built using various widely distributed features. Among those n-grams is basic and most commonly used feature set. Domain related information included in tweets is could also be used for classification of tweets. We here done experiments with 2 feature sets.

3.5.1 UNIGRAMS

In text classification most basic feature we can use is Unigrams. A tweet comprises of and is represented with multiple words present in that tweet. However, we have taken unigrams presence in our tweet as feature set. Rather than how many times a word is being repeated in sentence, its presence is very important in our consideration.

Pang et al. observed and concluded that unigrams gives us better output compared to repetition [4].This method will avoid scaling of data , thus in turn decreases time to train.

In our observation, Unigrams here follows Zipf's law. According to corpus present in natural language the word frequency is inversely proportional to frequency table rank .Data is well fitted as linear trendline.

f(frequency of word in tweet) α 1/r(rank in frequency table)

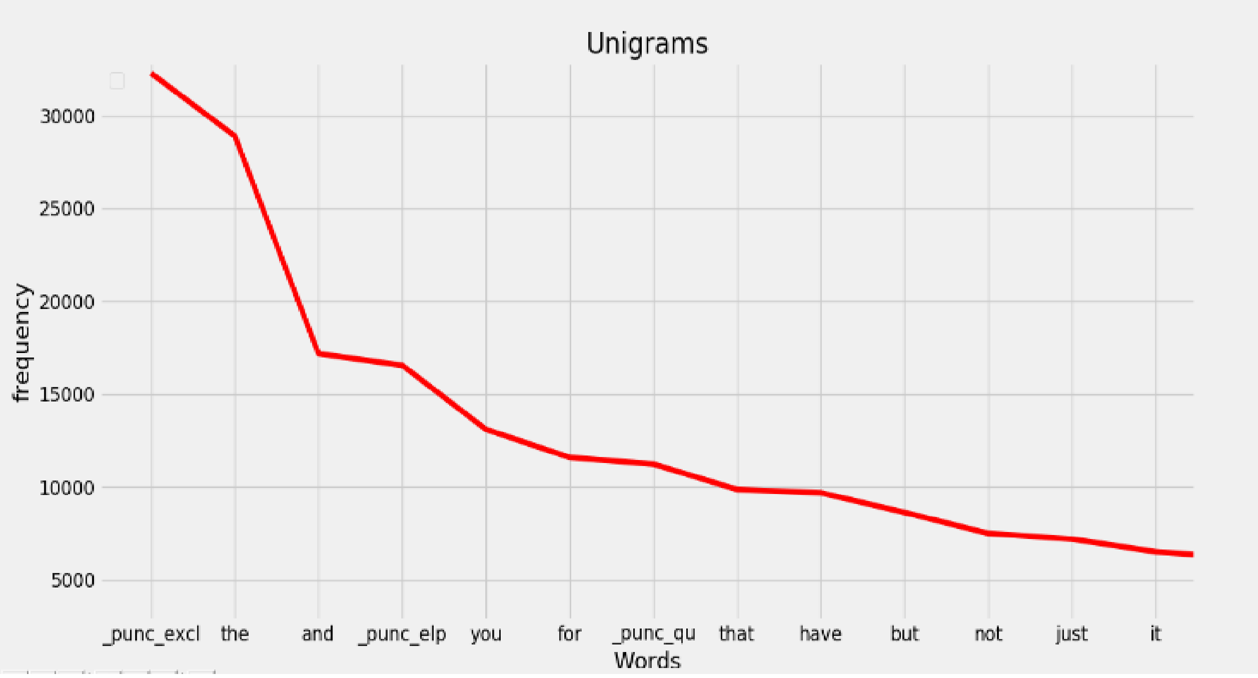


FIgure 3.2 most informative bigrams

3.5.2 N-GRAMS

N-grams generally means sequence of words of length-n .Prediction of next possible word given current context of a word can be possible with probabilistic model with usage of unigram ,bigram and trigrams . In sentiment analysis domain, n-grams performance is unclear. Pang et al. believes that usage of unigrams alone is better than using bigrams for example movie review classification. However, some other believe that trigrams and bigrams usage gives good results while classification of polarity for product reviews [4].

With increase in order of n-grams, they tend to be more sparse and sparse. In our experiment we have observed that with increase in number of tweets being considered number of tri and bigrams increases at rapid rate.

We here also choose to trim off n-grams whose frequency rate in corpus is less than one because of 2 reasons : 1)Sparse population of higher ordered n-grams 2)There might me chances that this n-grams are not good indicators of setiments .If we remove non-repeating n-grams, number of n-grams (i.e bi and trigrams) is considerably decreasing.

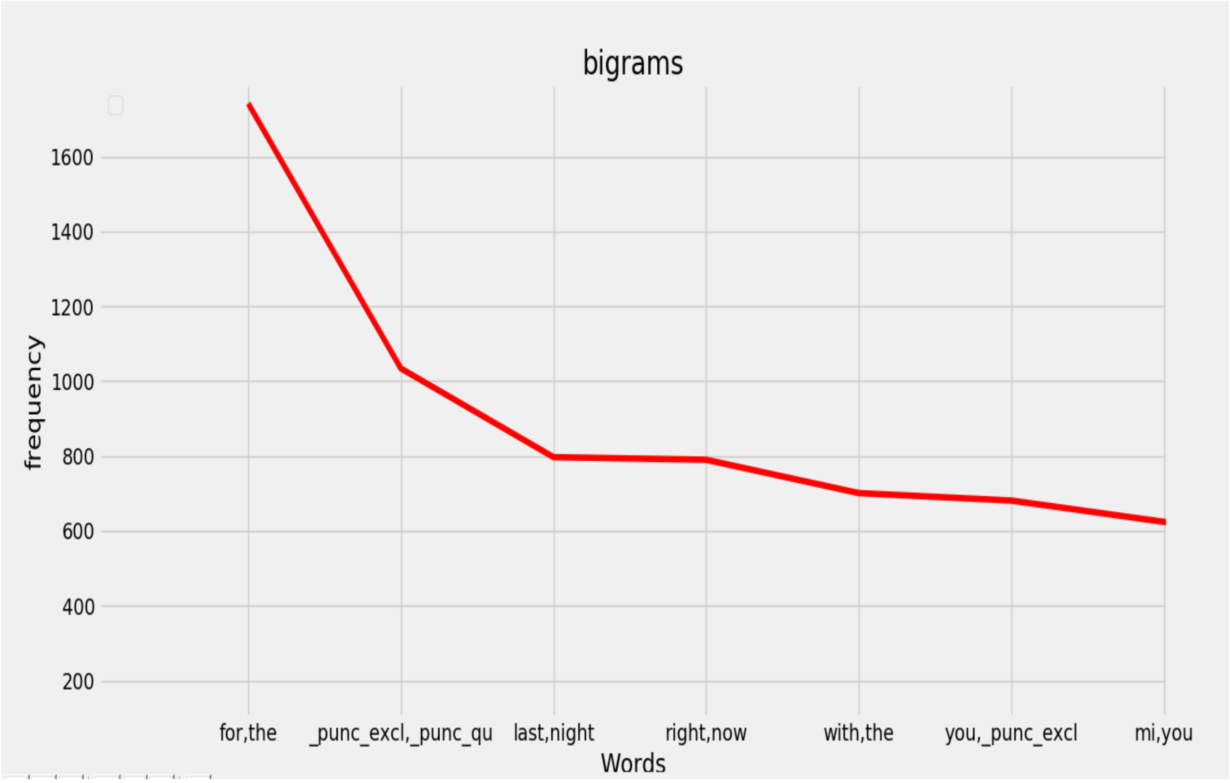


FIgure 3.3 most informative bigrams

3.5.3 NEGATION HANDLING

Sentiment analysis needs this negation detection which can be illustrated by difference in meaning such as, ”Something is good ” vs “Something is not good”. However, negatory words in natural language is comparatively simple. Negation Handling consists of 2 tasks – Explicit negation words detection in a sentence and these negation words scope on remaining words of a given sentence.”Council et al. made a look at whether negation words detection is useful in sentiment analysis of a sentence .One more factor is to consider here is upto what extant we are able to determine the scope of negation in text [10].They have described a method for detection of negation based on Left and Right Distances of a word in sentence to the nearest negation word.”

Explicit Negation Cues Detection :

Serial No	Negation Cues
1	Never
2	Noone
3	No
4	Nothing
5	Nowhere
6	Hadn’t
7	Haven’t
8	Hasn’t
9	Not
10	None
11	Wouldn’t
12	Wont
13	Couldn’t
14	Cant
15	Shouldn’t
16	Isn’t
17	Aint

18	Aren’t
19	Didn’t
20	Doesn’t
21	Don’t
22	Anywords with n’t as end.

Table 3.3 : List of Negation Words

SCOPE OF NEGATION:

Words which are nearer to the above mentioned negation words have more negative effect (hence considered as negative words) compared to words which are farther away from those words. Farthest words do not have any effect due to negation words. In this project we have found out left negativity (due to left placed negation word to the current word) and right negativity (due to right positioned negation word to the current word) to find out whether a particular word in sentence is in negative context of actual meaning or not.

Here place the example of scope of negation output that we have discussed.

Eg;-

1)He dont waste any time, he is not a bad guy.

2)Pavan and Charan are no one to do nothing.

Evaluate above examples using code and replace with output.

3.6 CLASSIFIERS

3.6.1 NAIVE BAYES CLASSIFICATION

It is the popular algorithm for test classification(it has proved it capability in many case).if we are have larger data points for training and smaller testing samples then in that scenes we opt for the naïve bayes classification. It works extensive fast and yields best results by using the probabilty concepts.

In easy terms “each features is independent of each other”. Even though they dependent on each other they at-last contribute independently for prediction. Naïve Bayes have outperformed many classification.

$$p(cl/z) = \; p(z/cl) * p(cl)/p(z) \tag{1}$$

$$p\left(\frac{cl}{z}\right) = p\left(\frac{z_1}{cl}\right) * p\left(\frac{z_2}{cl}\right) * p\left(\frac{z_3}{cl}\right) * \dots \dots . . * p\left(\frac{z_n}{cl}\right) * p(cl) \tag{2}$$

Terms in equation (1) & (2) are:

- Posterior probability of class(cl , tar) given predictor(z, attr) is P(cl/z).
- Prior probability of class is P(cl).
- P(z/cl) is the likelihood which is Probability of predictor given class.
- P(z) is the prior probability of predictor.

3.6.2 MAXIMUM ENTROPY CLASSIFIER

It is the popular text classifier, by parameterizing the model to achieve maximum categorical entropy, with the constraint that the resulting probability on the training data with the model being equal to the real distribution. It belongs to the class of exponential models. It is different from the Naïve Bayes Classifier Algorithm that it assumes that the features are conditionally independent from each other.

The main principle is that from all the models that fit our training data selects the one which has the highest entropy. Examples that this classifier can resolve are such as topic classification, language detection, sentimental analysis.

USAGE OF MAXENT CLASSIFIER:

We use this classifier when we don’t know about the prior distributions. This is used when we can’t use the conditional independence of the features. It requires more time to train than the Naïve Bayes Algorithm primarily due to optimisation problem .After computing these parameters it provides the robust results and it is competitive in terms of memory consumption and CPU.

MAXIMUM-ENTROPY BACKGROUND:

Unigrams, Bigrams and others are used to categorise it to a given class as positive, negative, neutral. Let there are n words $\{w_1, w_2, \ldots .w_n\}$ that can appear in the document. Then each document is represented by a sparse matrix 0's and 1's that indicate a particular word w_i exists in that document or not.

In Naive Bayes, the first step in building this model is take large amount of train data that includes samples in this specified format : (x_i, y_i) x_i is context based data which is sparse array and y_i represents its class. Second step we need to follow is to show train samples in terms of empirical probability distribution.

$$p(x, y) = 1/N * number\ of\ times\ that\ (x, y) occurs\ in\ the\ sample \tag{3}$$

Where N is the size of the training dataset.

Here the introduction of the following indicator function:

$$f_j(x, y) = \{\ 1\ if\ y = \ c_i\ and\ x\ contains\ w_k, 0\ otherwise \tag{4}$$

Generally, We call the above indicator function as "feature". In equation (3) & (4) only when particular class document is c_i and document has word w_k then indicator function whose values are nothing but binary returns 1.

3.6.3 SUPPORT VECTOR MACHINES

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N—the number of features) that distinctly classifies the data points. Here, we perform classification by finding the hyper-plane that differentiate the two classes very well.

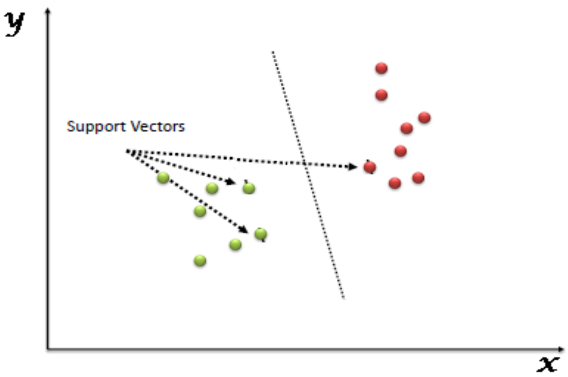
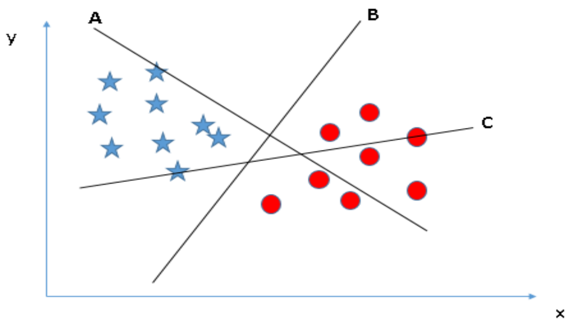
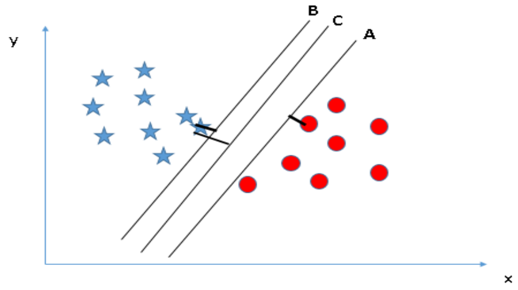
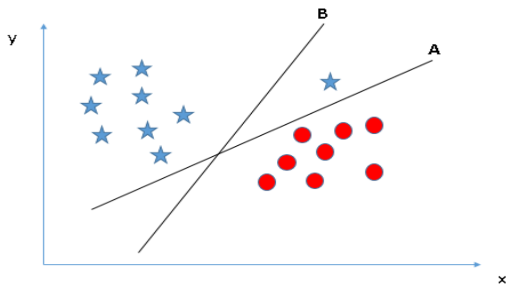
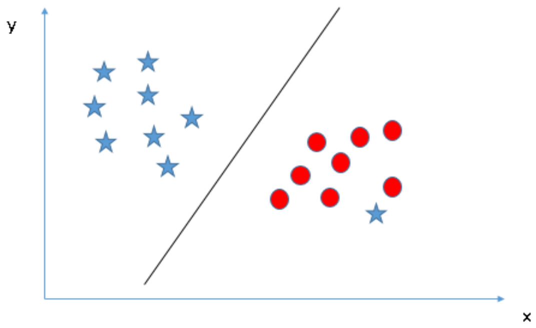


FIgure 3.4 Support Vector Machine (Classification of 2 sets of Data)

WORKING OF SVM:

 Figure 3.5: Scenario 1	 Figure 3.5: Scenario 2
 Figure 3.5: Scenario 3	 Figure 3.5: Scenario 4

Here in all figures x and y represents coordinate axis .

EXPLANATION:

In the above graphs scenario-1 indicates that there are three hyperplanes A,B,C,but among the three hyperplanes, hyperplane B acts as the best segregator to classify stars and circles.Scenario-2 indicates that hyperplane A,B are not chosen as the segregator because hyperplane C is at a maximum distance for both the data points (stars and circles).In Scenario-3 we choose hyperplane A as the classifier because if we choose hyperplane B a star is being classified into the circles group.Scenario-4 tells us that SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.

3.6.4 DECISION TREE

It can be learned by dividing the source nodes into subsets based on the attribute value test. This is repeatedly done in a recursive manner known as recursive partitioning. It is said to be completed when the subset of all the nodes have the same target value or by splitting no longer adds a value to predictions. The construction of a decision tree does not require any parameter setting and the domain knowledge. Decision trees have good accuracy. Decision tree is an induction approach to learn knowledge on classification.

Representation of a Decision Tree:

They classify the instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Instance is classified starting with root node of tree, attributes specified by each node is tested, then we move down to the corresponding branches on the basis of value of attribute. This entire process is repeated for subtree at each new node.

The decision tree learning algorithm:

The basic algorithmic rule employed in decision trees is thought because the ID3 (by Quinlan) algorithmic rule. The ID3 algorithmic rule builds decision trees employing a top-down, greedy approach. Briefly, the steps to the algorithmic rule are: - choose the most effective attribute → A - Assign A because the decision attribute (test case) for the NODE. - for every price of A, produce a replacement descendant of the NODE. - kind the coaching examples to the suitable descendant node leaf. - If examples ar absolutely classified, then STOP else retell over the new leaf nodes.

Now, consecutive huge question is a way to opt for the most effective attribute. For ID3, we expect of the most effective attribute in terms of that attribute has the foremost data gain, a live that expresses however well associate attribute splits that knowledge into teams supported classification.

Pseudocode: ID3 may be a greedy algorithmic rule that grows the tree top-down, at every node choosing the attribute that best classifies the native coaching examples. This method continues till the tree absolutely classifies the coaching examples or till all attributes are used.

Issues In Decision Trees:

Overfitting is avoided :- Since the ID3 algorithmic program continues ripping on attributes till either it classifies all the information points or there are not any additional attributes to splits on. As a result, it's liable to making decision trees that overfit by activity very well on the coaching knowledge at the expense of accuracy with relevancy the whole distribution of information.

There are, in general, 2 approaches to avoid this in decision trees: - enable the tree to grow till it overfits and so prune it. - stop the tree from growing too deep by stopping it before it utterly classifies the coaching knowledge.

A decision tree’s growth is per terms of the amount of layers, or depth, it’s allowed to own. Cross-validation can even be used as a part of this approach. Pruning the tree, on the opposite hand, involves testing the first tree against cropped versions of it. Leaf nodes are far from the tree as long because the cropped tree performs higher on the check knowledge than the larger tree.

Our initial definition of ID3 is restricted to attributes that fight a separate set of values. a way to create the ID3 algorithmic program additional helpful with continuous variables is to show them, in a way, into separate variables. Typically, whenever the classification changes from no to affirmative or affirmative to no, the common of the 2 temperatures is taken as a possible partition boundary.

CHAPTER 4 - EXPERIMENTATION

While performing sentimental analysis we tested with different classifiers (NaiveBayesClassifier, MaxentClassifier, DecisionTreeClassifier and SVMClassifier), different steps (1step, 2step), N-grams. we also tried to remove the impact of negative values on the sentiment.

We used sys [11] module of python for argument passing through command line.

Use the command line arguments this way:

```
>python sentimenttry.py logs/fileprefix
NaiveBayesClassifier,MaxentClassifier,DecisionTreeClassifier,SvmClassifier 1step,2step 1,3
0,1
```

We pick random 1 lakh data-points from Standford datsset and normalized for performing the sentimental analysis.

We will train and classify the normalized dataset totally 32 times according to the above command. Here classifiers are 4, methods are 2, n-grams are 2 and consideration of negativity is false or true , so total possible combinations are 4*2*2*2 = 32. Each time we do the training and classification, program automatically writes the output into file with given filename (timestamp). We rigorously perform all the combination and test to see which yield best result. This type of approach is hard because as it is likely to be as trial and error work to improve accuracy.

Steps we followed while training and classifying:

- If path of file where we need to write logs doesn't exist our code creates one .
- Classifier will be chosen according to the parameters passed.
- Choose the method(1step,2step)
- Follow the n-fold technique to train the dataset rigorously.
- For n times we get different training and testing dataset.
- Then we start training the classifier, check the accuracy, see the most informative features, plot the confusion matrix each time.
- By seeing the result of above we will try to tune the parameters of classifiers to get the best accuracy.

Steps for dividing the dataset into train and test sets in each case of n-fold technique:

- Before dividing, do a clear cut pre-processing work as mention above

- In each case complete dataset is divided into n parts, out of which n-1 parts are chosen as training set and 1 part is chosen as testing set
- Using frequency distribution we find unigrams, bigrams and trigrams accordingly.
- In feature Extraction We will try to get word features i.e has (word).
- In feature extraction we also get the negative features based on the scope of negation technique
- If method is 1step we do not consider objectivity of tweet but for 2step method we consider objectivity and sentiment of tweet and return 5 value(2 related to objective,2 related to sentiment,1 related test data)
- After getting train and test datasets through 1step method we perform classification, accuracy prediction, confusion matrix. And at last we plot all the accuracies of 10-foldes.
- If it is 2step we do classification, accuracy prediction, confusion matrix for both objective dataset and subjective dataset, then we plot graphs of accuracies.

CHAPTER 5 - RESULT

In this project, we also tested with negation detection for the purpose of sentiment analysis. We achieved around 80% accuracy for our baseline classifier as show below.

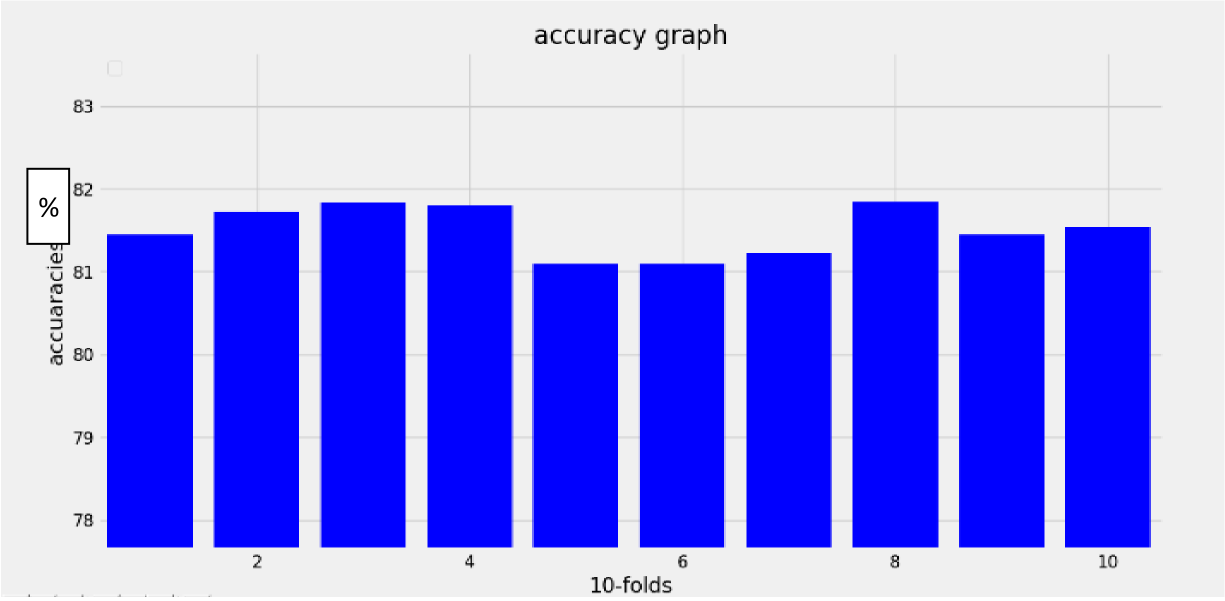


Figure 5. 1 : Accuracy graph Naïve Bayes classifier, 1step method unigrams, no negation

To increase the accuracy we tried to use the negation detection, unigrams, bigrams. if both n-grams and negation detection are used then accuracy decreases greatly.Figure 5.1 shows us graph between number of folds and accuracy percentage.

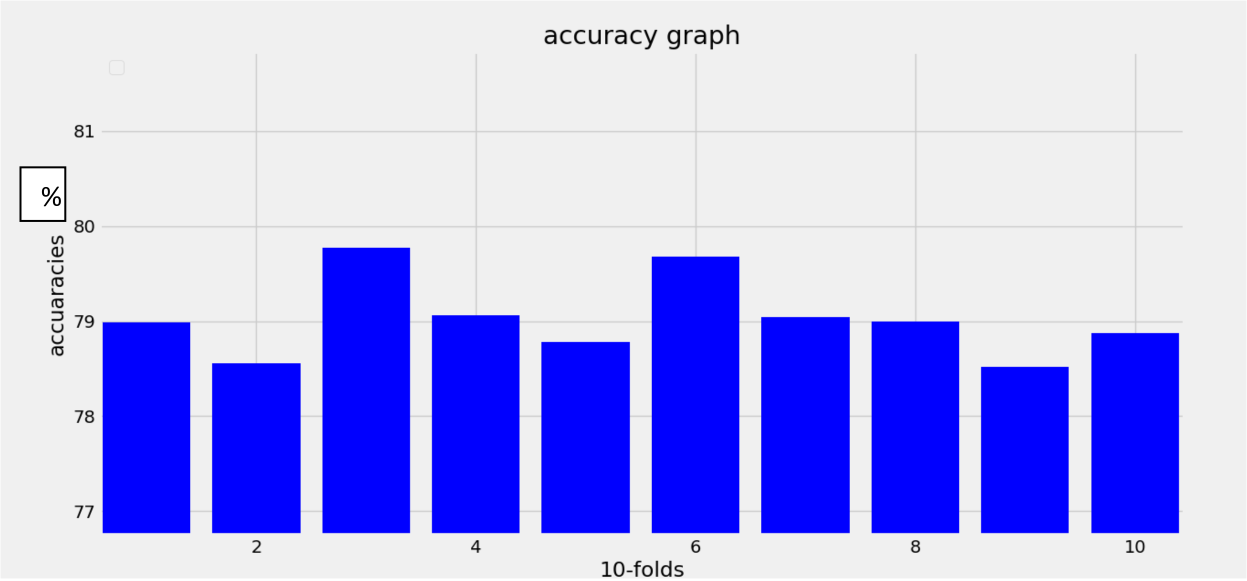


Figure 5. 2 :Accuracy graph for Naïve Bayes classifier, 1step method unigrams, negation

When compared to other classifiers NAЇVE BAYE's perform better than other.
Unigrams + bigrams+ trigrams obtained best accuracy when used with naïve BAYES classifier i.e 85.4%.

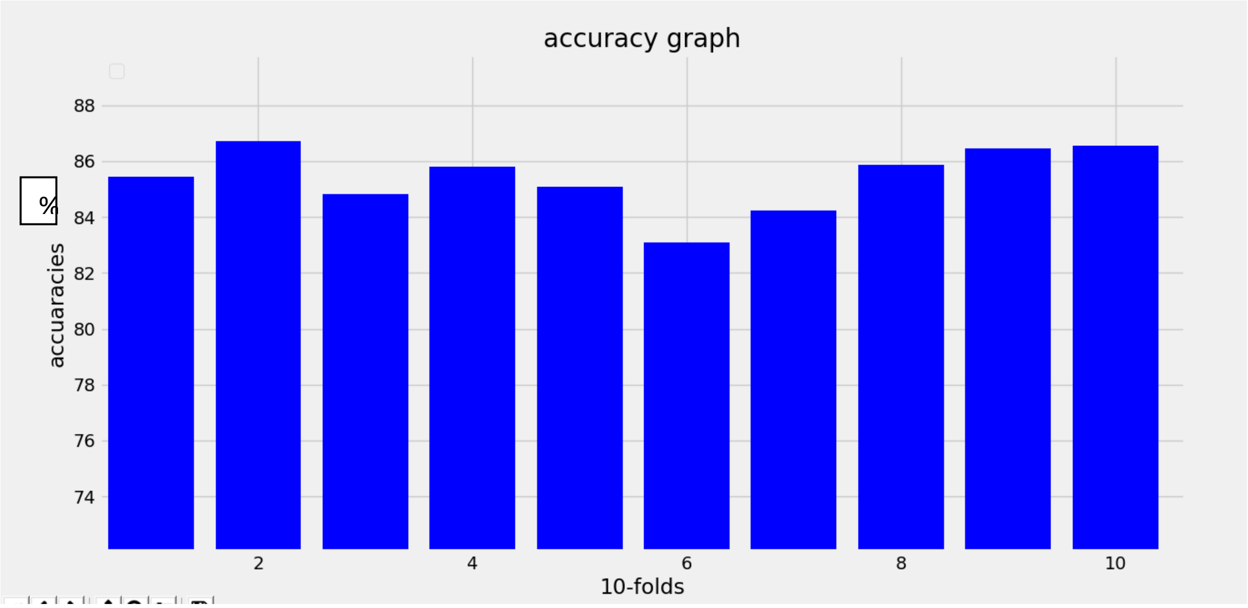


Figure 5. 3 : Accuracy graph for Naïve Bayes classifier, 1step method unigrams+ bi+ tri-grams, no negation

CHAPTER 6 - CONCLUSION AND FUTURE SCOPE

CONCLUSIONS:

Here we have a machine learning model which trains on publicly available Stanford corpus data and analyzes the sentiment of tweet (i.e , either tweet is negative or positive).We have used 3 different classifiers such as Naïve Bayes ,Maximum Entropy ,Decision Tree classifier to find out which classifier gives us best accuracy in terms of testing the data. We first get normalized tweets from Stanford corpus and the preprocess by replacing handles , URL's, hashtags and emoji's in the way so that we can train with classifiers. We here found out in our observation that concept of unigrams, bigrams, trigrams and negation considered given us best accuracy rather than ignoring them. Best accuracy of 85.4 percent is found when we use Naïve Bayes classifier,1step method, Uni and n-grams (Bi ,Tri grams) considered .So in our case we found out that same as many research papers already published comes to best among considered classifiers.

Even though Naïve Bayes classifier has provided us best accuracy there might be other options that we can consider such as SVM or Random forest so that we can get better accuracy than that we got in our case. future work will include best classifiers, methodologies so that we can get best accuracy than all the other works in this sentiment analysis field.

FUTURE SCOPE:

In future we would like to use the SVM for the above and see what accuracy variation occurs. We will also try to build a classifier for selecting a language tweets (like hindi, telugu etc).We also wanted to develop a module which takes user input which is sentence and can be able to analyze whether the input is negative or positive. Live Twitter data sentiment analysis also be added as module to our future work. So that we can also analyze what trends are being followed while user tweeting a tweet like statistics about number of average handles, emoji's and also predicting sentiment of next tweet user is about to tweet by analyzing the sentiment of his recent tweets.

REFERENCES

[1] B. P. a. L. Lee., "Opinion mining and sentiment analysis. Foundations and trends in information retrieval," pp. 1-135, 2008.

[2] E. K. a. E. L. Steven Bird, Natural language processing with python, United States of America.: O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2009.

[3] P. P. Alexander Pak, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," pp. 1320-1326, 2010.

[4] S. V. Bo Pang and Lillian Lee, "Thumbs up? Sentiment Classification using Machine Learning".

[5] S. L. a. D. M. P. Kushal Dave, "Mining the peanut gallery: opinion extraction," in *03: Proceedings of the 12th international conference on*, USA, 2003.

[6] J. F. Luciano Barbosa, "Robust Sentiment Detection on Twitter from Biased and Noisy Data".

[7] N. Sanders, "twitter-sentiment," [Online]. Available: http://www.sananalytics.com/lab/twitter-sentiment/.

[8] "Python Offical site," [Online]. Available: https://www.python.org/.

[9] "NUMPY(Python library)," [Online]. Available: http://www.numpy.org/.

[10] R. M. a. L. V. Isaac G Councill, "Negation and Speculation in natural language proccesing.," pp. 51-59, 2010.

[11] "SYS module of Python," [Online]. Available: https://docs.python.org/2/library/sys.html.