Sentiment Analysis of Movie Reviews Using Key Pair Graph Analysis

By

WASHIQ ANWAR SHAMSI
(153-35-1366)

A thesis submitted in partial fulfillment of the requirement for the degree

of Bachelor of Science in Software Engineering

Department of Software Engineering
DAFFODIL INTERNATIONALUNIVERSITY

Fall 2019

DECLARATION

It is to be hereby declared that this thesis has been done by me under the supervision of Mr. Asif Khan Shakir, Lecturer, Department of Software Engineering, Daffodil International University. It is additionally declared that neither this thesis nor any component of this has been submitted elsewhere for the award of any degree.

Washiq

Washiq Anwar Shamsi
Student ID: 153-35-1366

Batch**:** 18th
Department of Software Engineering
Faculty of Science & Information
Technology
Daffodil International University

Certified by:

Shakir
11.12.19

Asif Khan Shakir
Lecturer
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

ACKNOWLEDGEMENT

First of all, I am grateful to the Almighty Allah for giving me the facility to complete the final thesis.

I would relish expressing my gratitude to my supervisor Mr Asif Khan Shakir for the consistent avail of my thesis and research work, through his inspiration, energy, and erudition sharing. His effective direction assisted me to finding the solutions to research work and reach my final theory.

I would relish expressing my extreme sincere gratitude and appreciation to all of my edifiers of the Software Engineering Department for their kind assistance, benevolent advice and support during the study.

At last but not least, I would like to express my heartfelt gratitude to my beloved parents for their endurance, sacrifices, at most care, divine love and affection - without which I would have not succeeded !

Washiq Anwar Shamsi

\ \

TABLE OF CONTANTS

ABSTRACT

Sentiment Analysis is an automated mining of user generated opinionated text data such as reviews,comments and feedback.Sentiment Analysis classify those text data into their respective sentiments of positive , negative or neutral.Most of the researchers focused into this domain using one of the three classifier like SVM,Naive Bayes, and Maximum Entropy. In machine learning there are numbers of classifier model available.In this proposed approach there will be more focus on Mathematical Analysis and Natural Language Processing.The combinational difference between two subsets will provide the answer of movie review being positive or negative.In case of Natural Language Processing three algorithm has been used in this proposed model respectively Co_Occurrence matrix , Knowledge Graph Naive Bayes.To measure the combinational ratio of two subsets, Jaccard Distance has been used.Jaccard Distance is a pretty common technique in Mathematical and Big Data Analysis.In Feature Selection Jaccard Distance and Lexicon Based Approach has been used into proposed model.Co_Occurrence Matrix has been used to extract the feature selection.And to classify Knowledge Graph , Naive Bayes has been used.For determine the accuracy of the model "k-fold cross validation" has been performed.Keeping the value of k = 50.There are many researches on Naive Bayes Algorithm and Knowledge Graph Algorithm.But none of the researchers focused on the importance of merging these two techniques to perform Sentiment Analysis.This proposed model has shown how these two techniques can be merged together as a classifier.The co_occurrence frequency of each pair of words taken through Knowledge Graph. And occurrence frequency of each word is taken through Co_Occurrence Matrix.Combining the both co_occurrence and occurrence frequency have been taken to perform a probabilistic equation and traditional Naive Bayes Algorithm to measure the ratio of a context.The context results in two types of ratio positive and negative.If positive ratio is higher than the negative ratio the context will be positive else negative.Proposed Approach provides very comprehensive results on standard datasets.Out of two standard movie review data-sets, for one data-set proposed model outperformed all the previous result with accuracy of 88.56% and for other standard movie review data-set, it provides accuracy of91.82%.

Keywords—Co_Occurrence Matrix , jaccard Distance , Key Pair Graph ,
Knowledge Graph , Naive Bayes , Lexicon.

CHAPTER 1

INTRODUCTION

Sentiment can be defined as “A personal Positive or Negative feelings” [1], “Sentiment is a usually formulated as two class classification problem, positive and negative” [2]. With the rapid growth of the online discussion group, social network sites, and increased usage of the micro blogging there is the increase in the number of people providing their opinion online and labeling those sentiments can provide the great summaries to all those people who are looking forward to some advice or help from the online opinions [3].Sentiment Analysis is a process of mining on this user generated text content and determining the sentiment of users towards any particular thing like person, product or event and sentiments can be Positive, Negative or may be Neutral. Sentiment Analysis has become a very popular research area since 2000 [2]. After the research work published by [3] and in 2002, it really provided the very good directions to many of the researchers who are working in the domain of sentiment analysis. This domain is also known as the Opinion mining as well.This would be the first time starting from the internet era we are overwhelmed with a very huge volume of opinionated data over the social media sites and many other blogs, websites and This would be the first time starting from the internet era we volume of opinionated data over the social media sites and many other blogs, websites and forums, and without this data lot of research would not have been even possible. This led many of the researchers to focus on this area which is also having the huge potential for applications in many different areas. Opinions are always important to everyone whether it is individual,brands and services, governments or any other organization in the world, they play a very vital role in decision making. Business organizations are always in hurry to know that whether people like their products and services, what do people think about them, what kind of things people really like and don't like about their organization, product, service which may really help organizations to make decisions in a better way. Nowadays most of the people do not buy things without making some product analysis over the internet, people check for the product reviews and then make their decisions. Back in the time when organizations needed the public or consumers opinions they used to conduct the surveys and opinion polls which will require human resource and will be expensive as well as time consuming.

1.1 Background

Sentiment Analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. However, analysis of social media streams is usually restricted to just basic sentiment analysis and count based metrics. This is akin to just scratching the surface and missing out on those high value insights that are waiting to be discovered. So what should a brand do to capture that low hanging fruit? With the recent advances in deep learning, the ability of algorithms to analyze text has improved considerably. Creative use of advanced artificial intelligence techniques can be an effective tool fordoingin-depth research. We believe it is important to classify incoming customer conversation about a brand based on following lines:

1. Key aspects of a brand's product and service that customers care about.
2. Users underling interactions and reactions concerning those aspects.

1.1.1 Algorithms of Sentiment Analysis

1. Naive Bayes Classifier[6] - Naive Bayes Algorithm use to evaluate probabilistic equations through occurrence frequency of each independent words.It has one of the fast computation time than the other machine learning technique.

2. Maximum Entropy Classifier[7] - Maximum Entropy also evaluates probabilistic equations through entropy of the words or the entropy of combination sequence of words. Unlike the Naive Bayes Classifier its not assume the independent of each other.

3. Decision Tree[8] - Decision Tree is quite common process in every machine learning computation.In Sentiment Analysis Decision Tree has vast impact.First of all it measures the **“**Information Gain**”** against the targeted prediction.Than it maps a Tree based approach to find out all the possible result recursively. Because of the fact its searched for all possible answer its computation time becomes much slower than the other machine learning algorithm.But related to the fact it looks for all possible solution in some occasions it provides better accuracy than the other machine learning algorithms.

4. Support Vector Machine[9] - Support Vector Machine use for classification and regression model.It uses the labeled data of training set.Basically it creates multiple class in a two dimensional graph with the data points which creates from train set.With Hyperplane it divides the classes.While testing it maps the test set from the graph and provides the result.

5. Lexicon Based Approach[10] - Its a dictionary based approach.It takes words from the test set and scores the positive and negative words.If in a sentence positive score is higher than the negative then the sentence is positive else negative.There are many techniques for scoring the words.

1.1.2 Levels of Sentiment Analysis

Sentiment analysis can mainly be carried out on any of the following two levels, known as document level sentiment analysis and sentence level sentiment analysis.

Document Level: This focuses on classifying the whole document into its respective sentiments of positive, negative or neutral. Movie reviews or product reviews generally fall into this category. Most of the previous research in the sentiment analysis focused on document level only and many of them worked on the movie reviews only. This paper focuses on sentiment analysis of movie reviews which is nothing but the document classification, more information about the movie review dataset and experiments are provided in section 4: Experiments and Result Analysis.

Sentence Level: Other approach is known as sentence level sentiment analysis in which only sentences are going to be analyzed and then will be classified as positive and negative polarity of a sentence. Sentence level sentiment analysis is being very popular nowadays because of the popularity of micro-blogging sites such as twitter and many other, which deals with short sentences which are limited to only 140 characters, and also influence many researchers to work on this sort ofplatforms.

1.1.3 Technique of Sentiment Analysis

There are mainly two methods to carry out the sentiment analysis, first is known as Supervised approach or Machine Learning based approach which make use of machine learning classification techniques and other is known as Unsupervised or Lexicon based approach, which is also known as dictionary based approach.

Supervised Learning: In supervised learning test data or unclassified data is going to be classified based on the data available in the training dataset.Training dataset is the one which is already labeled and uses the classifier algorithm to classify new data based on the labeled data or training data. Number of classifier algorithms like Support Vector Machine (SVM), Naive Bayes and Maximum Entropy are mostly used classifier algorithm to carryout sentiment analysis. [3] were the first one to use the concept of supervised learning classifier in the area of sentiment analysis, they worked with above mentioned three classifiers used the concept of Unigrams for the feature selection and they found that Support vector Machine performs better compared to other classifiers. Reason to use above mentioned three classifier is because they work greatly in the area of text classification . More details on these approaches are discussed in Chapter 2- Literature Review.

Unsupervised Learning: Lexicon based approach is also known as the dictionary based approach or semantic based approach. This approach do not require separate training and testing dataset but instead of that list of words or dictionary of words will be used to classify the text data in form of sentence or document. Much of the research based on lexicon approach make use of available lexical resources such as dictionary of positive and negative words which are going to be used to classify the sentence or document. As if there are some positive words in the sentence then it means that sentence represents positive polarity, and if there are negative words then it represents negative polarity of a sentence or document. This concept was first started by [4]. More details on these approaches are discussed in Chapter 2- LiteratureReview.

1.1.4 Previous Work on Sentiment Analysis of Movie Reviews

For previous model[11] accuracy are achieved for two dataset using random forest by manually changing values of all three different hyperparameters. Both the dataset are evaluated on train and test split by keeping the ratio of 80% for training and 20% for Testing.Dataset[12] V1.0 was first converted into word vector of 10000 words with removing stopwords, after this Information Gain was applied for feature reduction with threshold value of 0.002. This resulted in total of 2275 features. Random forest classifier with hyperparameters values for number of trees 900, number of features at random 12, depth value was set to unlimited and that provided classification **accuracy of 87.85%**. Dataset V2.0 was first converted into word vector of 10000 words with removing stop words, after this Gain Ratio was applied for feature reduction with threshold value of 0.00. This resulted in total of 1942 features. Random forest classifier with hyperparameters values for number of trees 400, number of features at random 11, depth value was set to unlimited and that provided classification **accuracy of 91.00%**. There number of different values of each hyperparameter are tried for both the datasets and above mentioned values of hyperparameters are the ones that provided good results. Though different values of hyperparameters are tried manually for each iteration and based on the accuracy returned in that iteration hyperparameters values will be updated for the next iteration. However we have mainly focused on two hyperparameter that is number of trees and number of features. In which increase in number of trees linearly increases accuracy up to certain values and after that there will not be any drastic change in accuracy results.

1.2 Motivation of the Research

Now a days astronomically immense dataset are available in gregarious network,e- commerce site,product reviews etc. Its pretty easy now to perform a sentiment analysis and ascertain the presage that can be positive or negative or equivalent. So , there are lot of ways to implement this “Research Work” into the process of NLP[13].In this Era of globalization and internet , there is no lack of “Specific Dataset for Sentiment Analysis” -which is the most fascinating part of the proposed research.Many researchers had done lot of researches to gain more precision as much as possible on Sentiment Analysis. So researching on “Dataset Analysis” has kick commenced an acute competition that how long our proposed model can go compared to other research[11] results towards hundred percent perfection.To achieve such perfection we have been utilizing a concrete Dataset that’s been antecedently utilized by otherresearchers.

1.3 Problem Statement

This paper focuses on experimental evaluation on two standard Movie Review datasets. Dataset is available at [12], which is usually conceded as the gold standard data set for the researchers working in the domain of the Sentiment Analysis. First Dataset is known as Movie Review Dataset V1.0 which consist of 1400 movie review out of which 700 reviews are positive and 700 reviews are negative. Second dataset consist of total 2000 Movie reviews and 1000 of which are positive and 1000 of which are negative. The main reason for using this data set is that, they are already classified in to the two classes which are Positive and Negative, so all the reviews which are positive by their contextual sentiments they are kept in to the positive directory and the one that are negative by their contextual sentiment are kept in the negative directory. All those reviews are in the text file format. Lots of researchers has done work on this dataset and gained promising accuracy , the target is build a proposed approach that can also provide a approximate result like other research.

1.4 Research Questions

3. Question 1: Make a propose model that can decide whether themovie review is positive or negative?
4. Question 2: Merge multiple machine learning algorithmto get accurate probabilistic result.
5. Question 3: Make a comparable ratio between two sentences.

1.5 Research Objectives

This research is based on Natural Language Processing[13] , this proposed model will gain more accurate result than other NLP[13] based model on this specific dataset.

Make a suitable technique for using two different types of features selection such as Jaccard Distance and Subset of Words Showing how to Marge this two feature and normalize a huge textual dataset.

Develop a technique that can handle Naive Bayes and Knowledge Graph Algorithm in the same time in reasonable time complexity.

1.6 Research Scope

Sentiment Analysis become quite common now a days.Lot of dataset are available to perform sentiment analysis.It can be use for depression analysis by using social networks dataset , it can use to predict the effectiveness of a product from product reviews and also can use to medical issues to find out the cause and treatment of a mental health diseases and more.

1.7 Thesis Organization

This paper includes five sections: Introduction, Literature Review, Research Methodology, Result and Discussion, and Conclusions. Introduction section discuss about the research background, research objective, problem statement, research question and research scope.Literature review section discuss about the related work of this research and research gap. Research methodology section, shown a proposed model for the research and discuss about the research methodology. Result and Discussion section, shown the result of the methodology with discussion. Finally, Conclusion section, discuss the final output of the result and future recommendations.

CHAPTER 2

LITERATURE REVIEW

To understand Sentiment Analysis properly , its quite important to learn what is sentiment analysis and how it works.It possible to gather knowledge about Sentiment Analysis and some important techniques to implement Sentiment Analysis [14].To perform Textual Analysis (Zhang)[22] , Sentiment Analysis is commonly used.Sentiment Analysis is a process which calculates the sentence polarity and subjectivity. In the process of Sentiment Analysis many Machine Learning Algorithm might be used (Md.Rafiqul Islam et al)[8].Sentiment Analysis has lot of scope in real life[23,24,25].In the present era users put review for products,foods,opinion etc.There are many possible ways to use this data.A research done by (Md.Rafiqul Islam et al)[8] has shown how to predict user has depression or not with twitters data-set using Sentiment Analysis.The problem and cure of a patient with mental issue has been figured out , just taking only an interview of the patient by using Knowledge Graph Algorithm (Morihito Takita et al) [15].Knowledge Graph is a model of Sentiment Analysis.Movie Review can also be defined as "Product Review".Many researches researched on Sentiment Analysis[14] using the Movie Review data set to find out whether the review is positive or negative.

Because of the fact this proposed method is based on NLP[13]. Its also important to gain knowledge about NLP[13] for ensure to maintain proper structure that follows the NLP(Chowdhury,G.G)[26] methods.Statistical Processing of NLP has the technique to pre-process the data quite perfectly."Statistical Processing of NLP" goes through two phases during pre-processing. These are : Data Preprocessing and Parameterization.

Data Preprocessing is a way to collect Key Words from a large context.It has three steps Elimination of Tags , Standardization , Stemming and Lemmatization . Usually lot of special characters are used in real life communication , all of these special characters are not important for Analysis so most of the characters get removed except keywords in the step of Elimination of Tags.After removing the tags the data get Standardized by considering its context.Contextcanbe different in every issue.Basically in textual analysis the text gets splitted by each word.Sometime the words get Standardized against its co_occurrence , occurrence , length , polarity ratio , subjective ratio etc. After completion of Standardization , Stemming and Lemmatization process starts.In Stemming phase , the Bese Words get memoized.For example: **word: “Comput”** has four different **“{er,es,e,ing}”** forms.The word or the token reduced into most common form and pushed into the memoization.Its not important to get a dictionary based word it can be out of dictionary as well.In this case word **“comput”** will get memoized.The most attractive part of NLP is that it can be used to analyze all languages not only one or two languages.In Lemmatization stage the words those has the dictionary form after getting removed the derivatives and svaed into Lexicon which is created from train data-set.

Test data-set remains as raw data-set.As a result it needs to be processed again.Yet the pre-processing of test data set can be performed using the Lexicon which was created in Stemming and Lemmatization phase.This process is known as Parameterization.

Proposed method pre-process the data on the basis of Lexicons (E.Riloff et al) [10] and Subset of words. There are several works that can show how Lexicon[25] can be implemented and can give a precise result to ensure accuracy.Lexicon is a dictionary that creates from classifier.It takes each word from classifier and store as a Noun , Verb , Adjective , Adverb etc.This process can also be said as “Bag of Words Classifier[27]”.While creating the Lexicon the Stop Word from sentence get removed. For Example : **“The Movie Was Great ! ”** in this sentence stop words are **{The , Move}** . Reason for saying these words as “Stop Words” because these don’t take a huge part while analyzing a data.After removing the unnecessary words from a sentence , remaining words are stored into a Bag of Words Classifier or in a Lexicon.The context get decided as a positive , negative or neutral context against the subjectivity ratio and the polarity ratio from a word . This approach does not give an expected accuracy but this is one of the ways that the result can be decided.In case of Feature Selection and other Algorithms such as (Naive Bayes[25], Knowledge Graph[15], Word2vec[16] etc) creation of Lexicon or “Bag of Words[27]” are quite common.

Jaccard Similarity (Ivchenko, G. I., & Honov, S. A)[29] is basically used to measure the similarity between two sets and Jaccard Distance (Shameem, M.-U.-S., & Ferdous, R)[28] is used to measure the dissimilarity between two sets.Both the techniques use a set-based formula in mathematics to find out the combination ratio between two strings or two contexts. Jaccard similarity and Jaccard Distance compare the members of two sets to see which members are shared and which distinct.These measure the two set data between the ratio of 0 to 1. The steps of measuring the Jaccard similarity between two sets or contexts are, 1) Calculating the Union ratio between two sets 2) Calculating the Intersection ratio between two sets 3) Dividing the ratio(2) by ratio(1). And then to measure Jaccard Distance[28] Ratio, Jaccard Similarity[29] should be subtracted from One (1). In the case of Jaccard Similarity the higher the ratio the more similarity contains the compared set.

Co_Occurrence matrix (Gotlieb) [18] is such matrix which takes words or tokens as a row and column.This process can be called as text vectorization(Liu, S., Fan, X., & Chai, J.)[30,31].Every dimension of a matrix indicates the sentence.By comparing the dimension of a matrix sentence subjectivity can be measure.But just looking at the occurrences from each indices from the matrix will not provide a good result.Cause it will break the order sequence from a sentence.So for this reason “n gram” solution has come.n = 1 means token , n = 2 means token pairs.Its also possible to compare the pair of words by watching pair frequency of both words from the matrix and then measure the subjective ratio from a sentence. “n-gram[32]” solution will provide a good accuracy for Textual Analysis.”n-gram[32]” is a very common technique in NLP.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This approach is focused on Natural Language Processing[13] and some mathematical concept. Especially on sets, combinations and probability. The probabilistic approach in this methodology is taken from the Naive Bayes Algorithm[6].Naive Bayes Classifier uses far less computing power compared to other methods and often is a baseline method for many models. A Naive Bayes Classifier is a probabilistic machine learning model that is used for the classification task.

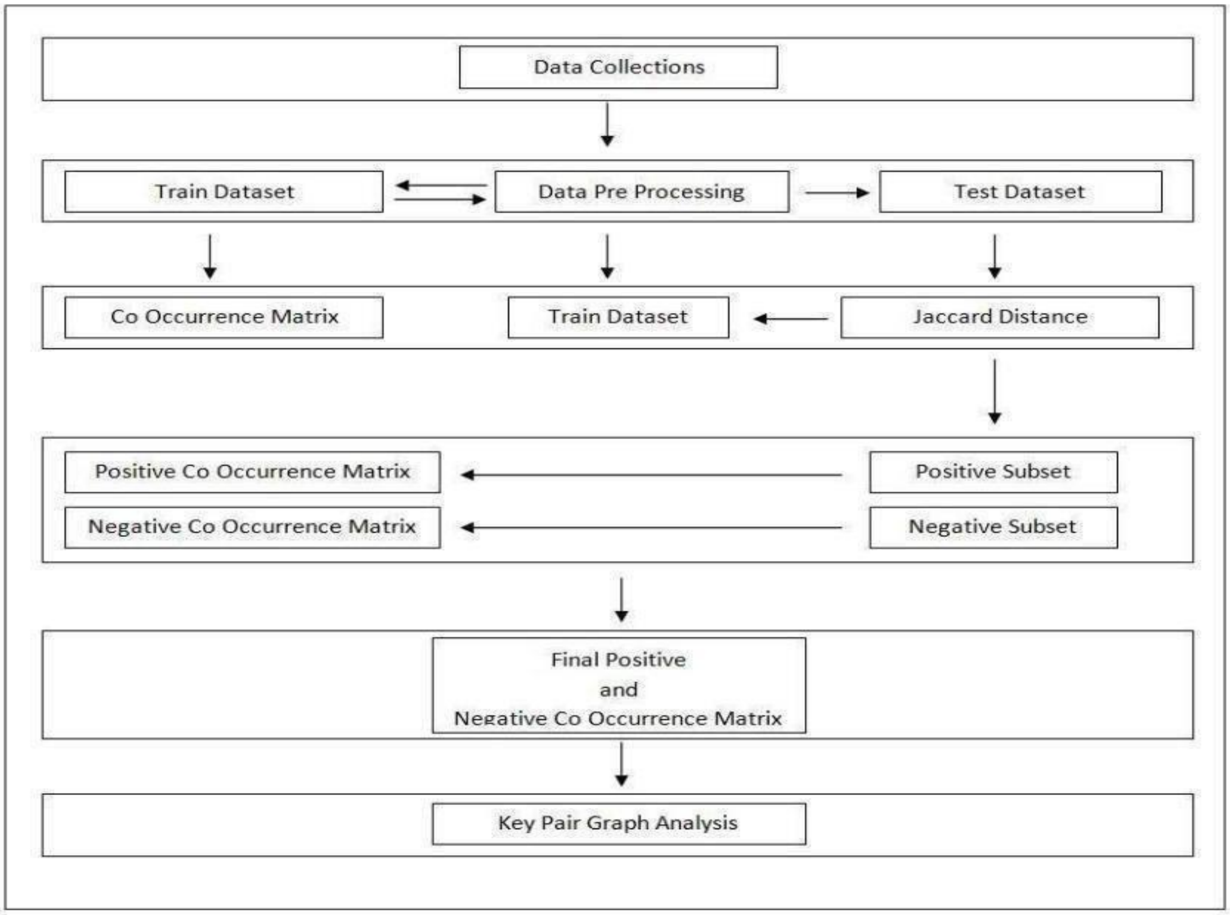


Figure 3.1: Proposed model for Sentiment Analysis of Movie Reviews

After pre-processing the datasets the Co_Occurrence matrix[18] has been created. Using Jaccard Distance[17] most probable subsets has been selected and those subsets have also been pushed into the Co_Occurrence matrix[18]. After getting the final Co_occurrence Matrix[20] the Key Pair Graph has been analyzed. The idea of a Key Pair Graph has come from Knowledge Graph[21]. In many occasion of NLP[13], Knowledge Graph[15] is generating many important decisions.

The Co_Occurrence Matrix has created with a set of words. These words have pre-processed with Lexicon based approach more described in section (3.3). The words are kept as row and column in Co_Occurrence Matrix. Each row and column has a specific occurrence frequency. And the indices of each row and column is the frequency of both pair of words.Co_Occurrence Matrix used to extract the feature. With these frequencies, two classifier algorithm has been performed Naive Bayes And Knowledge Graph. From Knowledge Graph in each iteration, the total count of co_occurred pair has figured out from test classifier more described in section (3.6). Because of the fact, this Knowledge Graph is taking action with each co_occurred pair of word so it named a Key Pair Graph Analysis. With the occurrence frequency of each row and column, Naive Bayes Classifier has performed.

Formula of Naive Bayes for measuring positive and negative ratio from test set -

For Positive Occurrence Ratio from Positive Co_Occurrence Matrix:

1. $P\_NBR = PCF[row]/(PCF[row]+NCF[row])$
2. $P\_NBC = PCF[col]/(PCF[col]+NCF[col])$

For Negative Occurrence Ratio from Negative Co_Occurrence Matrix:

1. $N\_NBR = NCF[row]/(PCF[row]+NCF[row])$
2. $N\_NBC = NCF[col]/(PCF[col]+NCF[col])$
3.

Where $P\_NBR =$ Positive Naive BayesRow , $P\_NBC =$ Positive Naive Bayes Column , $PCF[row] =$ Positive Occurrence Frequency of Row , $PCF[col] =$ Positive Occurrence Frequency ofColumn , $NCF[row] =$ Negative Occurrence Frequency ofRow , $NCF[col] =$ Negative Occurrence Frequency of Column, $N\_NBR =$ Negative Naive Bayes Row , $N\_NBC =$ Negative Naive Bayes Column.

Actually, from the test set, two types of Co_Occurrence matrix has generated Negative Co_Occurrence Matrix and Positive Co_Occurrence Matrix. Positive Co_Occurrence Matrix is providing positive ratio and Negative Co_Occurrence Matrix is providing Negative ratio. In Traditional Co_Occurrence Matrix, it compares the dimension of the matrix against the test set and provides the subjectivity or the polarity ratio of the test set. But in the proposed model rather comparing the dimension it's comparing the ratio between the matrix column and matrix row against the test set with the help of Naive Bayes and Knowledge Graph Classifier. Above described formula performs the probabilistic equations of traditional Naive Bayes Algorithm from both positive and negative Co_Occurrence matrix and also provides positive and negative ratio. Bellow mentioned formulas show how these two algorithms can be merged with probabilistic mathematics.

Positive Ratio Measuring

1. $PPR = PCF[row][col]/PCF[row]$
2. $PPC = PCF[row][col]/ PCF[col]$
3. $Pair = PCF[row][col]$
4. $PP = \sum(((PPR + PPC) / 2 + Pair) + (P\_NBR * P\_NBC))$ [Adding Naive Bayes Ratio]
5. $PTP = \sum PTP + i$
6. $PR = PP*(PTP / Total\_Positive\_Words)$

Negative Ratio Measuring

1. $NPR = NCF[row][col]/NCF[row]$
2. $NPC = NCF[row][col]/ NCF[col]$
3. $Pair = NCF[row][col]$
4. $NP = \sum(((NPR + NPC) / 2 + Pair) + (N\_NBR * N\_NBC))$ [Adding Naive Bayes Ratio]
5. $NTP = \sum NTP + i$
6. $NR = NP*(NTP /Total\_Neagtive\_Words)$

Where PPR = Positive Probable Row , PPC = Positive Probable Column , $PCF[row][col]$ = Positive Occurrence Frequency from the index of Row and and Column.

PP = Positive Probability of Pair, PTP = Positive Total Pair , PR = Positive Ratio.

NPR = Negative Probable Row , NPC = Negative Probable Column , $NCF[row][col]$ = Negative Occurrence Frequency from the index of Row and and Column.

NP = Negative Probability of Pair, NTP = Negative Total Pair , NR = Negative Ratio.

After Adding the Naive Bayes Ratio with total summation of connected pairs frequencies probabilistic ratio two type ratio has measured from test set , Positive Ratio and Negative Ratio.If Positive Ratio is higher than the Negative Ratio than the test set is positive else negative.

In proposed model **“**k fold cross validation**”** has been used.where k value set to 50.Described in chapter (4).

3.2 Data Collection

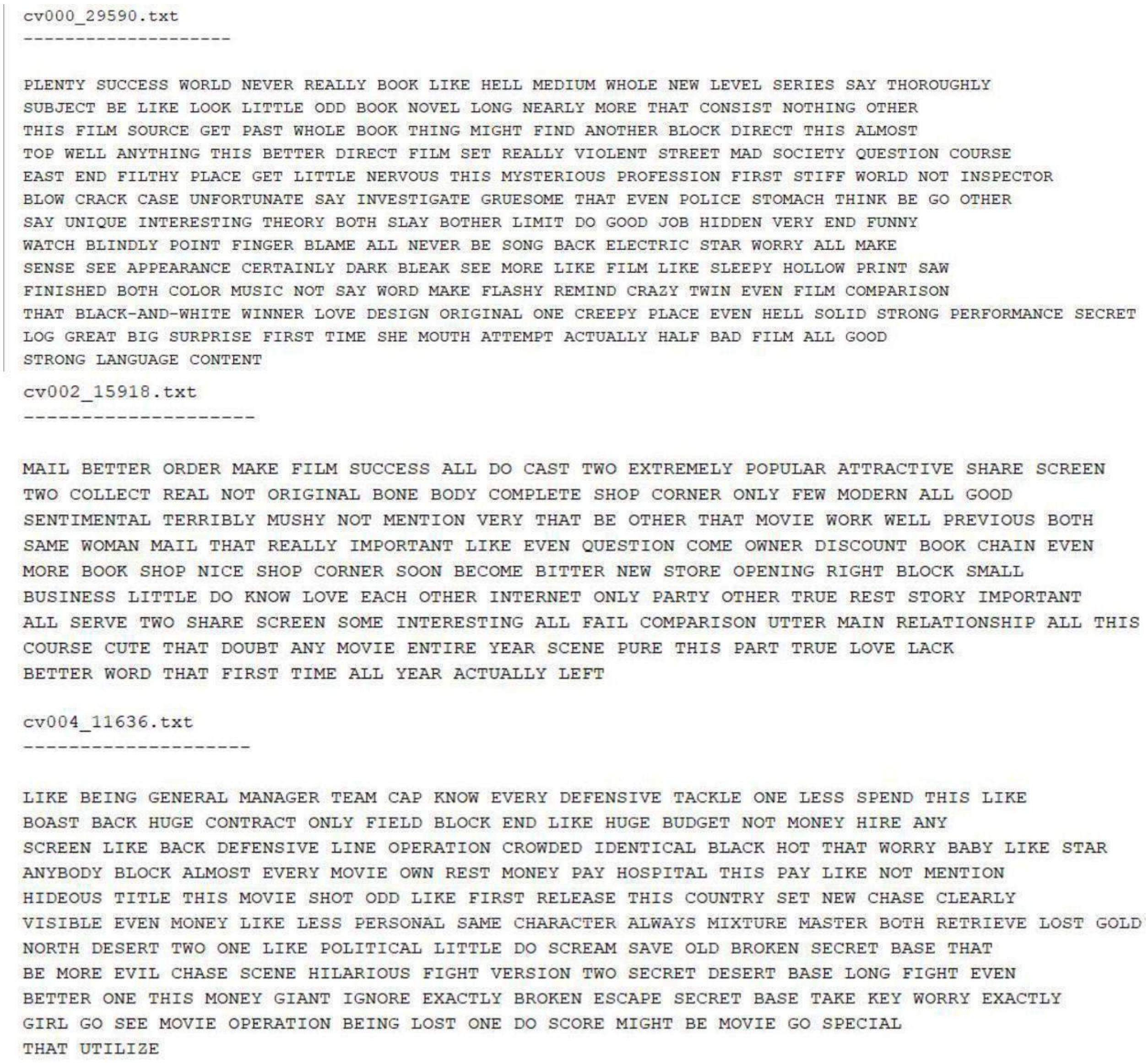
Datasets is collected from imdb movie reviews[12]. More importantly lots of researcher has worked on this datasets. Most helpful thing on this datasets is that , the positive and negative datasets is already defined. So that it will be more easier to train the data and test the data. There are two versions of datasets. Each review contains in a single document.Version one contains 700 positive documents and 700 negative documents.Version two contains 1000 positive documents and 1000 negative documents.

3.3 Data Preprocessing

First of all from the the train dataset ,each reviews were minimized to a small pieces of sentence. Only the noun , verb , adverb , adjective and higher occurred words has been taken from each reviews. So that each review became a set of words or set of tokens. With these tokenizer a lexicon created from classifier **(fig 3.3)**. On the first view of analysis it seemed only the noun , adjective , verb , adverb wasn’t enough to decide the proper result. From version one dataset first 560 reviews were trained and last 140 reviews were tested as sample case. From that sample test case it assumes that word length more than five and occurred more than 30 times in Train Dataset can be added to the set of tokens , so that the result can be more promising.So for the other train dataset occurrence of words depends on word_length > 5 and the ratio of 30/560. For Example : if the sample traindataset had 800 train reviews than word appear ratio would be **“**(30/560)*800 & word_length > 5**”**.Algorithm for creating each review as set of token mentioned bellow :-

```
String[] words = Document[i].Splits(“ ”);
$if(words[i])$ is adjective ,noun ,verb or adverb
$elseif(words[i]\_occurrence > (30/560)*Document.Length)$
$\Pr e\Pr ocessDocumnet[i].Add(words[i])$
```

Figure 3.2: After PreProcessing Train Dataset
Each Review Created as set of tokens



```
cv000_29590.txt
--------------------

PLENTY SUCCESS WORLD NEVER REALLY BOOK LIKE HELL MEDIUM WHOLE NEW LEVEL SERIES SAY THOROUGHLY
SUBJECT BE LIKE LOOK LITTLE ODD BOOK NOVEL LONG NEARLY MORE THAT CONSIST NOTHING OTHER
THIS FILM SOURCE GET PAST WHOLE BOOK THING MIGHT FIND ANOTHER BLOCK DIRECT THIS ALMOST
TOP WELL ANYTHING THIS BETTER DIRECT FILM SET REALLY VIOLENT STREET MAD SOCIETY QUESTION COURSE
EAST END FILTHY PLACE GET LITTLE NERVOUS THIS MYSTERIOUS PROFESSION FIRST STIFF WORLD NOT INSPECTOR
BLOW CRACK CASE UNFORTUNATE SAY INVESTIGATE GRUESOME THAT EVEN POLICE STOMACH THINK BE GO OTHER
SAY UNIQUE INTERESTING THEORY BOTH SLAY BOTHER LIMIT DO GOOD JOB HIDDEN VERY END FUNNY
WATCH BLINDLY POINT FINGER BLAME ALL NEVER BE SONG BACK ELECTRIC STAR WORRY ALL MAKE
SENSE SEE APPEARANCE CERTAINLY DARK BLEAK SEE MORE LIKE FILM LIKE SLEEPY HOLLOW PRINT SAW
FINISHED BOTH COLOR MUSIC NOT SAY WORD MAKE FLASHY REMIND CRAZY TWIN EVEN FILM COMPARISON
THAT BLACK-AND-WHITE WINNER LOVE DESIGN ORIGINAL ONE CREEPY PLACE EVEN HELL SOLID STRONG PERFORMANCE SECRET
LOG GREAT BIG SURPRISE FIRST TIME SHE MOUTH ATTEMPT ACTUALLY HALF BAD FILM ALL GOOD
STRONG LANGUAGE CONTENT
cv002_15918.txt
--------------------

MAIL BETTER ORDER MAKE FILM SUCCESS ALL DO CAST TWO EXTREMELY POPULAR ATTRACTIVE SHARE SCREEN
TWO COLLECT REAL NOT ORIGINAL BONE BODY COMPLETE SHOP CORNER ONLY FEW MODERN ALL GOOD
SENTIMENTAL TERRIBLY MUSHY NOT MENTION VERY THAT BE OTHER THAT MOVIE WORK WELL PREVIOUS BOTH
SAME WOMAN MAIL THAT REALLY IMPORTANT LIKE EVEN QUESTION COME OWNER DISCOUNT BOOK CHAIN EVEN
MORE BOOK SHOP NICE SHOP CORNER SOON BECOME BITTER NEW STORE OPENING RIGHT BLOCK SMALL
BUSINESS LITTLE DO KNOW LOVE EACH OTHER INTERNET ONLY PARTY OTHER TRUE REST STORY IMPORTANT
ALL SERVE TWO SHARE SCREEN SOME INTERESTING ALL FAIL COMPARISON UTTER MAIN RELATIONSHIP ALL THIS
COURSE CUTE THAT DOUBT ANY MOVIE ENTIRE YEAR SCENE PURE THIS PART TRUE LOVE LACK
BETTER WORD THAT FIRST TIME ALL YEAR ACTUALLY LEFT

cv004_11636.txt
--------------------

LIKE BEING GENERAL MANAGER TEAM CAP KNOW EVERY DEFENSIVE TACKLE ONE LESS SPEND THIS LIKE
BOAST BACK HUGE CONTRACT ONLY FIELD BLOCK END LIKE HUGE BUDGET NOT MONEY HIRE ANY
SCREEN LIKE BACK DEFENSIVE LINE OPERATION CROWDED IDENTICAL BLACK HOT THAT WORRY BABY LIKE STAR
ANYBODY BLOCK ALMOST EVERY MOVIE OWN REST MONEY PAY HOSPITAL THIS PAY LIKE NOT MENTION
HIDEOUS TITLE THIS MOVIE SHOT ODD LIKE FIRST RELEASE THIS COUNTRY SET NEW CHASE CLEARLY
VISIBLE EVEN MONEY LIKE LESS PERSONAL SAME CHARACTER ALWAYS MIXTURE MASTER BOTH RETRIEVE LOST GOLD
NORTH DESERT TWO ONE LIKE POLITICAL LITTLE DO SCREAM SAVE OLD BROKEN SECRET BASE THAT
BE MORE EVIL CHASE SCENE HILARIOUS FIGHT VERSION TWO SECRET DESERT BASE LONG FIGHT EVEN
BETTER ONE THIS MONEY GIANT IGNORE EXACTLY BROKEN ESCAPE SECRET BASE TAKE KEY WORRY EXACTLY
GIRL GO SEE MOVIE OPERATION BEING LOST ONE DO SCORE MIGHT BE MOVIE GO SPECIAL
THAT UTILIZE
```

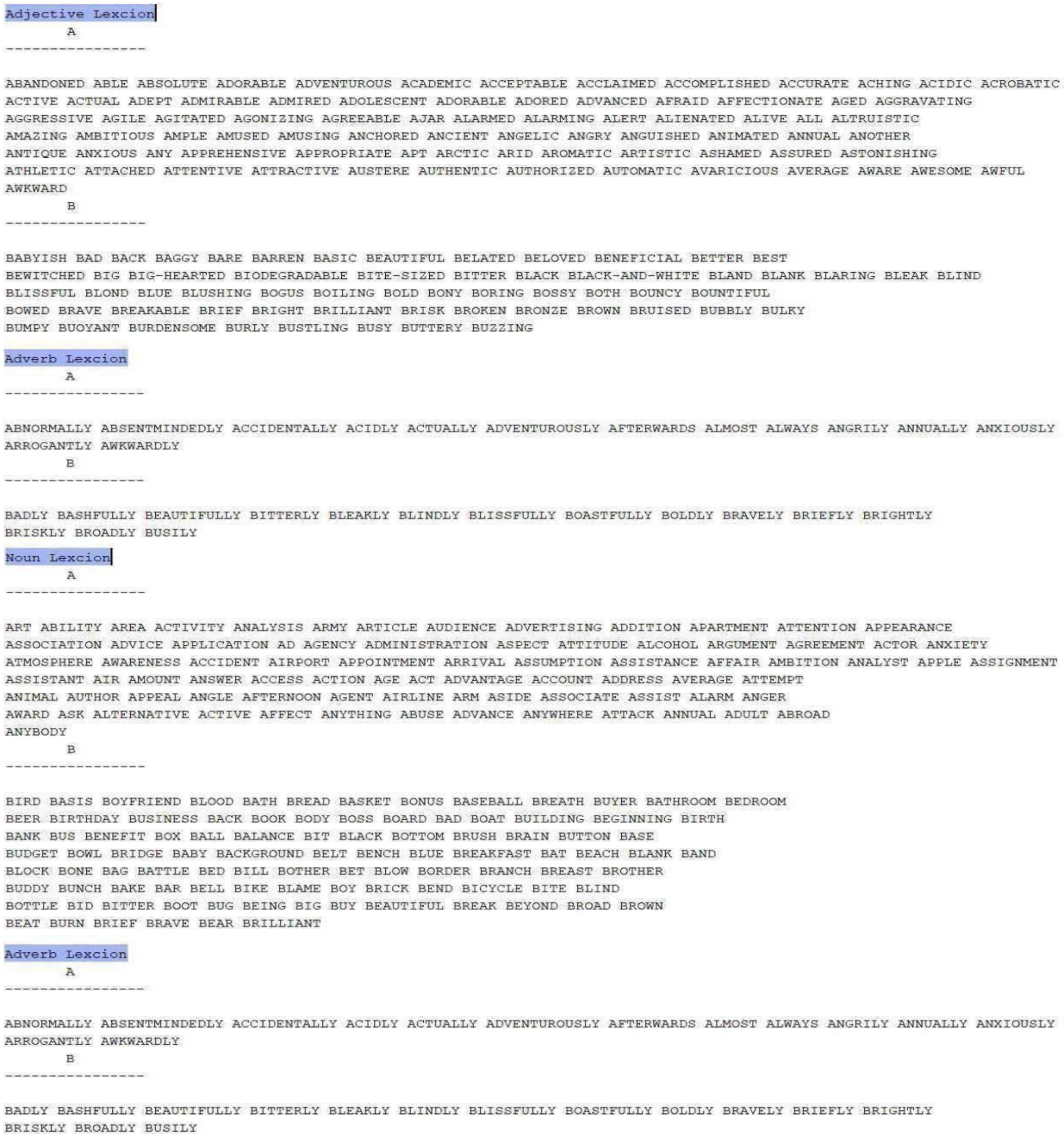
“Each space separated word a pieces tokens”

While selecting the tokens from the train documents , all the tokens where memorized with a unique id storing to the HashMap Data Structure. While preprocessing the test documents the token were selected from the HashMap which was created while processing the train dataset.

3.3.1 Creating Lexicons

Lexicon set were created of words from pre processed classifier ,with these set of Lexicons both the co_occurrence matrix[18] generated positive and Negative.More importantly with these Lexicon set , test data has pre processed so , there had no chance to get new tokens from test classifier. showed in **fig(3.3)**. The words occurred more than the ratio of **“**30/560**”** and words those are from adjective,noun,verb and adverb were added to the Lexicon list.Lexicon is nothing but a dictionary that has been created from test set.These Lexicon set used as a feature selection in this proposed model.By creating the Co_Occurrence matrix feature selection has extracted.Rather taking words from the traditional dictionary creating a manual dictionary from train set will make computation time more faster while searching the words.Its obvious that the words needs to get searched for the test set.Lexicon Set has been created from A to Z but in the diagram **fig(3.3)** only A and B sets has shown.

Figure 3.3: Lexicon Set starts with A and B from Classifier.(Each word is space spectated)



```
Adjective Lexcion
       A
----------------

ABANDONED ABLE ABSOLUTE ADORABLE ADVENTUROUS ACADEMIC ACCEPTABLE ACCLAIMED ACCOMPLISHED ACCURATE ACHING ACIDIC ACROBATIC
ACTIVE ACTUAL ADEPT ADMIRABLE ADMIRED ADOLESCENT ADORABLE ADORED ADVANCED AFRAID AFFECTIONATE AGED AGGRAVATING
AGGRESSIVE AGILE AGITATED AGONIZING AGREEABLE AJAR ALARMED ALARMING ALERT ALIENATED ALIVE ALL ALTRUISTIC
AMAZING AMBITIOUS AMPLE AMUSED AMUSING ANCHORED ANCIENT ANGELIC ANGRY ANGUISHED ANIMATED ANNUAL ANOTHER
ANTIQUE ANXIOUS ANY APPREHENSIVE APPROPRIATE APT ARCTIC ARID AROMATIC ARTISTIC ASHAMED ASSURED ASTONISHING
ATHLETIC ATTACHED ATTENTIVE ATTRACTIVE AUSTERE AUTHENTIC AUTHORIZED AUTOMATIC AVARICIOUS AVERAGE AWARE AWESOME AWFUL
AWKWARD
       B
----------------

BABYISH BAD BACK BAGGY BARE BARREN BASIC BEAUTIFUL BELATED BELOVED BENEFICIAL BETTER BEST
BEWITCHED BIG BIG-HEARTED BIODEGRADABLE BITE-SIZED BITTER BLACK BLACK-AND-WHITE BLAND BLANK BLARING BLEAK BLIND
BLISSFUL BLOND BLUE BLUSHING BOGUS BOILING BOLD BONY BORING BOSSY BOTH BOUNCY BOUNTIFUL
BOWED BRAVE BREAKABLE BRIEF BRIGHT BRILLIANT BRISK BROKEN BRONZE BROWN BRUISED BUBBLY BULKY
BUMPY BUOYANT BURDENSOME BURLY BUSTLING BUSY BUTTERY BUZZING

Adverb Lexcion
       A
----------------

ABNORMALLY ABSENTMINDEDLY ACCIDENTALLY ACIDLY ACTUALLY ADVENTUROUSLY AFTERWARDS ALMOST ALWAYS ANGRILY ANNUALLY ANXIOUSLY
ARROGANTLY AWKWARDLY
       B
----------------

BADLY BASHFULLY BEAUTIFULLY BITTERLY BLEAKLY BLINDLY BLISSFULLY BOASTFULLY BOLDLY BRAVELY BRIEFLY BRIGHTLY
BRISKLY BROADLY BUSILY

Noun Lexcion
       A
----------------

ART ABILITY AREA ACTIVITY ANALYSIS ARMY ARTICLE AUDIENCE ADVERTISING ADDITION APARTMENT ATTENTION APPEARANCE
ASSOCIATION ADVICE APPLICATION AD AGENCY ADMINISTRATION ASPECT ATTITUDE ALCOHOL ARGUMENT AGREEMENT ACTOR ANXIETY
ATMOSPHERE AWARENESS ACCIDENT AIRPORT APPOINTMENT ARRIVAL ASSUMPTION ASSISTANCE AFFAIR AMBITION ANALYST APPLE ASSIGNMENT
ASSISTANT AIR AMOUNT ANSWER ACCESS ACTION AGE ACT ADVANTAGE ACCOUNT ADDRESS AVERAGE ATTEMPT
ANIMAL AUTHOR APPEAL ANGLE AFTERNOON AGENT AIRLINE ARM ASIDE ASSOCIATE ASSIST ALARM ANGER
AWARD ASK ALTERNATIVE ACTIVE AFFECT ANYTHING ABUSE ADVANCE ANYWHERE ATTACK ANNUAL ADULT ABROAD
ANYBODY
       B
----------------

BIRD BASIS BOYFRIEND BLOOD BATH BREAD BASKET BONUS BASEBALL BREATH BUYER BATHROOM BEDROOM
BEER BIRTHDAY BUSINESS BACK BOOK BODY BOSS BOARD BAD BOAT BUILDING BEGINNING BIRTH
BANK BUS BENEFIT BOX BALL BALANCE BIT BLACK BOTTOM BRUSH BRAIN BUTTON BASE
BUDGET BOWL BRIDGE BABY BACKGROUND BELT BENCH BLUE BREAKFAST BAT BEACH BLANK BAND
BLOCK BONE BAG BATTLE BED BILL BOTHER BET BLOW BORDER BRANCH BREAST BROTHER
BUDDY BUNCH BAKE BAR BELL BIKE BLAME BOY BRICK BEND BICYCLE BITE BLIND
BOTTLE BID BITTER BOOT BUG BEING BIG BUY BEAUTIFUL BREAK BEYOND BROAD BROWN
BEAT BURN BRIEF BRAVE BEAR BRILLIANT

Adverb Lexcion
       A
----------------

ABNORMALLY ABSENTMINDEDLY ACCIDENTALLY ACIDLY ACTUALLY ADVENTUROUSLY AFTERWARDS ALMOST ALWAYS ANGRILY ANNUALLY ANXIOUSLY
ARROGANTLY AWKWARDLY
       B
----------------

BADLY BASHFULLY BEAUTIFULLY BITTERLY BLEAKLY BLINDLY BLISSFULLY BOASTFULLY BOLDLY BRAVELY BRIEFLY BRIGHTLY
BRISKLY BROADLY BUSILY
```

3.4 Co_Occurence of Matrix

Each Specific document from train dataset were created as a set of Lexicons or subset of words. Two different co_occurrence matrix[18] were created **Fig(3.8.1.1,3.8.1.2)** from train dataset.Positive co_occurrence matrix[18]and negative co_occurrence matrix[18].From **Fig(3.8.1.1,3.8.1.2)** Both space separated word is row and column And indices are co_occurrence of both words. All the Tokens were stored in the Lexicon in training stage.The row and columns of the matrix is the tokens.Total occurrence of each tokens also kept on track for the Key Pair Graph analysis.The indices of the matrix is count of the total pair of both tokens , it means that how many times both tokens occurred as a pairs.In **Fig(3.8.1.1,3.8.1.2)** white value indicates both tokens arrived number as a pair.Tokens are in column and row and their total occurrence inside of closing bracket. Black values are initial to zero because those row and column didn't made any pair in test set.

3.5 Jaccard Distance

Jaccard Distance[17] is a mathematical form that used to find out the distance between to sets.

Formula of Jaccard Distance:

if A,B is a Set than **"**J(A,B)= (($A \bigcup B$) - ($A \bigcap B$)) /($A \bigcup B$) **".**

For Example:

if set A={1,2,3,4} and set B = {3,4,5,6}

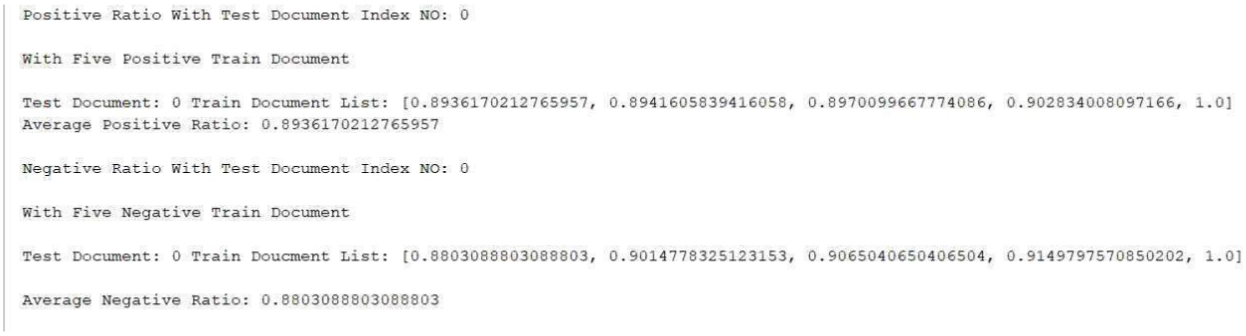
$A \bigcup B$ ={1,2,3,4,5,6} = 6

$A \bigcap B$ ={3,4}=2

($A \bigcup B$)-($A \bigcap B$) =6-2=4

J(A,B)=4/6=0.66, Distance between two set is 0.66.

While pre processing the document **(Fig 3.2)** each train documents were created on the basis of set of tokens. And all the test documents also created in same fashion. So its quite possible to perform jaccard distance[17] approach.Previously mentioned 80% dataset used for training and 20% dataset used for testing. In the testing stage each and every documents which belongs to 20% test data were compared with 80% train data though Jaccard Distance[17] **(Algorithm2)**.Each test document will contain a specific value against each train document. This values are the distance between both set of tokens. All of this distance values where stored into TreeSet Data Structure , so that the distance values can store in acceding order.



```
Positive Ratio With Test Document Index NO: 0

With Five Positive Train Document

Test Document: 0 Train Document List: [0.8936170212765957, 0.8941605839416058, 0.8970099667774086, 0.902834008097166, 1.0]
Average Positive Ratio: 0.8936170212765957

Negative Ratio With Test Document Index NO: 0

With Five Negative Train Document

Test Document: 0 Train Doucment List: [0.8803088803088803, 0.9014778325123153, 0.9065040650406504, 0.9149797570850202, 1.0]

Average Negative Ratio: 0.8803088803088803
```

Fig 3.5.1: Performing Jaccard Distance Formula
With First Test Index with Train Data-Set
First Five Jaccard Ratio is Provided

From Fig(3.5.1) its seems that first test document average jaccard positive distance ratio is 0.89 and average negative distance ratio is 0.88. So the test index document **"Index 0"** has a less possibility to being positive.

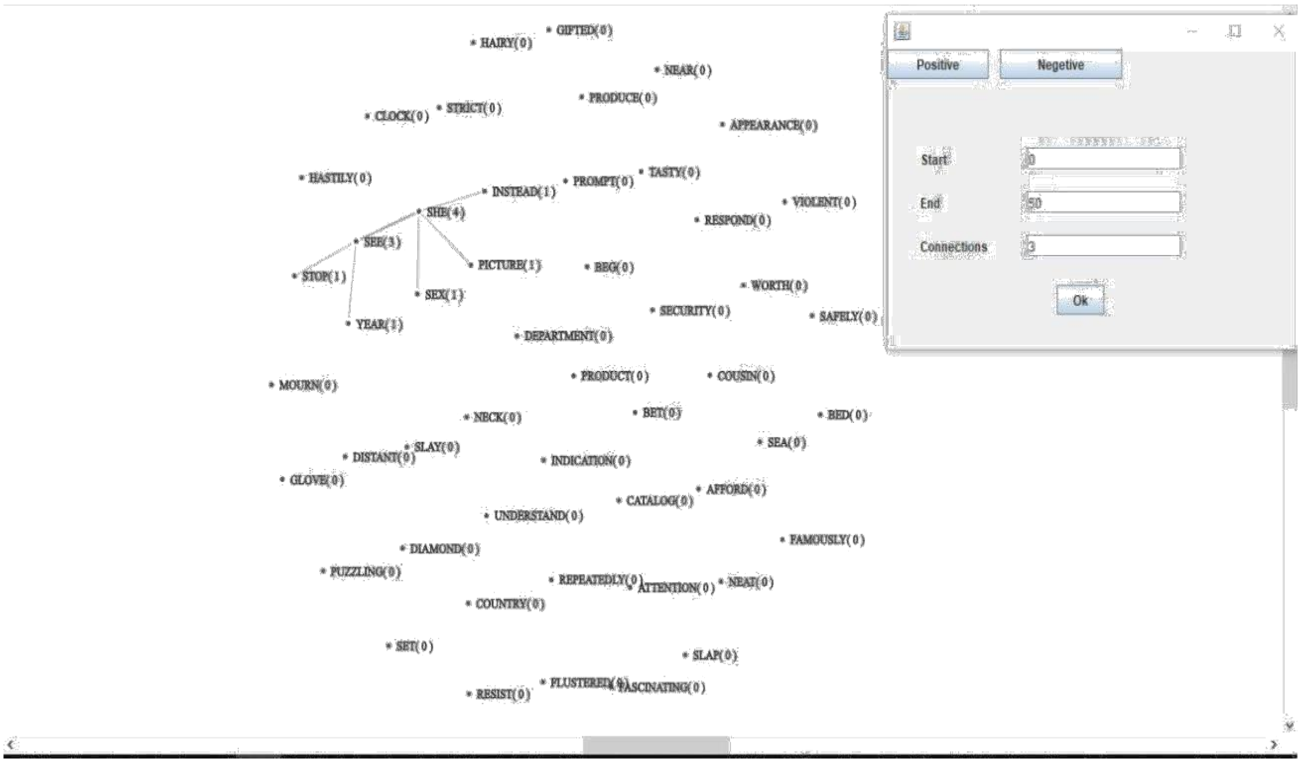
$$T \in Positive = J\,(\mathrm{Pre\_Pr}ocessed\ \_TestData[i]\ ,\ \mathrm{Pre\_Pr}ocessed\_\ DocumentPositive[])$$
$$T\ \in Negative\ =\ J(\mathrm{Pre\_Pr}ocessed\_TestData[i]\ ,\ \mathrm{Pre\_Pr}ocessed\_DocumentNegative[])$$
Where T=TreeSet;

Formula 3.5.1: Formula of Getting Jaccard Distance Ratio
From Classifier

Each Test Document will compare though Jaccard Distance with Each Positive Test Document and Negative Test Document.After calculating average of first 100 values from both TreeSet , if the differences between two average point is equal or grater than 0.005 than the particular test document also has been pushed to the co_occurrence matrix[18] **(Fig 3.8.1.1 , 3.8.1.2)**. The Algorithm is described on Appendix A.Analyzing the sample test case , it clarifies that keeping the difference ratio to 0.005 gives the most significant probable positive and negative subset from the test dataset.

3.6 Key Graph Analysis

Key pair graph analysis , this idea has generated from knowledge graph algorithm[15].From the co_occurrence matrix[18] **(Fig 3.6.1,3.6.2)** the occurrence of each token was taken to calculate Naive Bayes Algorithm[6]. And the pair amount " *CM* [*i*][*j*]; Co_Occurrence Matrix of ith Row and jth Column" of both tokens were taken to calculate probabilistic math.Merging both Naive Bayes[6] and probabilistic math the final result has been generated.**From (Algorithm3,4)** If Positive_Ratio[i] > Negative_Ratio[i] than Document[i] will grant as positive ,else negative.In **(Fig 3.6.1)** clarifies that on behalf of first 50 tokens and minimum pair connection is 3 If **{'Instead','she','see','stop','year','picture'}** these words or tokens in a sentence has concurrent pair connection with each other than the sentence has high chance to being positive against this classifier.



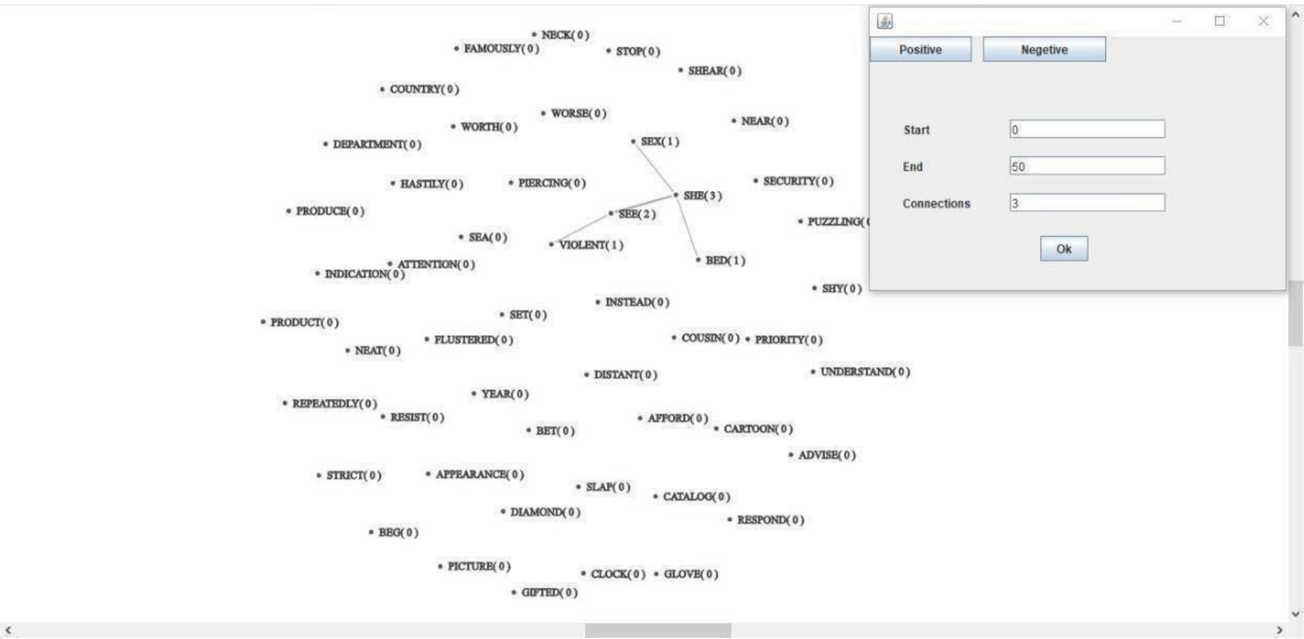
Start 0

End 50

Connections 3

Fig 3.6.1: Positive KeyGraph Of First 50 Tokens Generated from Positive Co_Occurrence Matrix

In (Fig 3.6.2) also clarifies that on behalf of first 50 tokens and minimum pair connection is 3 if **{‘she’,’see’,’violent’,’bed’}** these words or tokens in a sentence has concurrent pair connection with each other than the sentence has high chance to being negative against this classifier.



Start 0

End 50

Connections 3

Fig 3.6.2: Negative KeyGraph Of First 50 Tokens Generated from Negative Co_Occurrence Matrix

3.7 Experiment With Feature And Approach

Previously described , there were two standard movie review dataset[12]. Version 1 Dataset contains 1400 movie review from there 700 data is positive and 700 data is negative. Version 2 Contains 2000 movie reviews from there 1000 data is positive and 1000 negative.Previously mentioned approaches from proposed model few used for feature selection and few used for classifications.

Table 3.7.1 : Experiment Table

DataSet Versior	Approach	Feature Selections	**Accuracy**
1	Knowledge Graph,Naive Bayes	Jaccard Distance,Lexicons	**88.56%**
2	Knowledge Graph,Naive Bayes	Jaccard Distance,Lexicons	**91.82%**

3.7.1 Comparison With Previous Models

Comparison of Data-Set V1

Serial No	Author	Approach	Feature Selection	Accuracy %
1	Pang and Lee[3]	Naive bayes SVM, Maximum Entropy	Unigram, Bigram	82.90%
2	Mullen and Collier[9]	Support Vector Mechine	Unigramsyntatic relations	86%
3	E. Riloff et al[10]	Lexicon Based Approach	Unigram Biagram	82.70%
4	Xue Bai[17]	Two Stage Markov Blanket Classifier	All words and theii Subset	87.52%
5	Hitesh Parmar and GloryShah[11]	Random Forest	Unigrams	87.85%
6	Proposed Model	Knowledge Graph,Naive Bayes	Jaccard Distance,Lexicons	88.56%

Sr No	Author	Approach	Feature Selection	Accuracy %
1	Pang and Lee[33]	Naive Bayes SVM	Graph Based Approach	87.20%
2	Kennedy & Inkpen[34]	SVM	Unigrams,Bigrams	85.90%
3	Ruj Xia[36]	Naive Bayes,SVM, MaximumEntropy	Unigram , Bigrams Dependency Grammar	86.40%
4	Zhu Jian[35]	Back Propagation	Unigram	86.00%
5	Agarwal and Mital[37]	SVM	Unigram+Rough Set Theory	87.60%
6	Prabowo et.al.[38]	ID3 , SVM	Document Frequency	89.00%
7	Sharma and Dey[39]	NB,SVM,ME,DT	Unigrams	90.90%
8	Konig and Brill[40]	Hybrid Approach	N-grams	91.00%
9	Xue Bai[17]	Two Stage Markov Blanket Classifier	All words and subset of words	92.00%
10	Hitesh Parmar and Glory Shah[11]	Random Forest	Unigrams	91.00%
11	A.Abbasi et.al.[5]	SVM	Hybrid Feature Selection of Information Gain + Genetic Algorithm	95.55%
12	Proposed Model	Knowledge Graph,Naive Bayes	Jaccard Distance,Lexicons	91.82%

Comparison of Data-Set V2

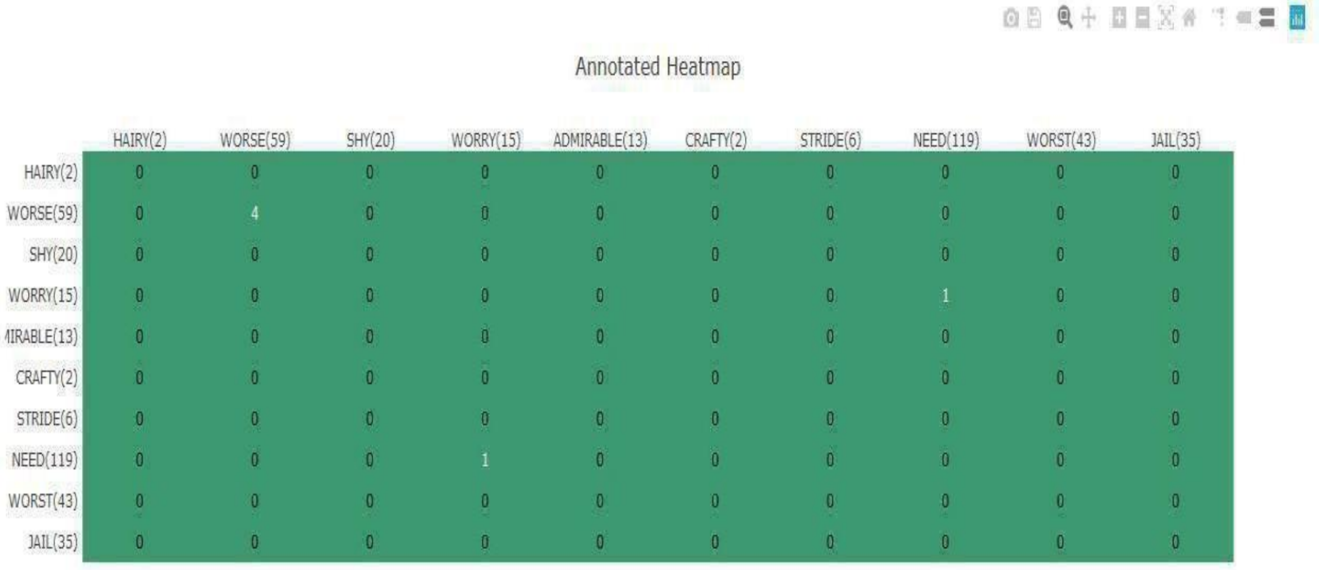
The Comparison chart is taken from [11]

3.8 Visualization

Two visualization has been shown.Knowledge Graph and Co_Occurrence matrix. The visualization was created using JavaScript , Java and Json files. The visualization API was taken from[20,21].This tools has the ability to visualize large amount of object and the object can change its view by every input.

3.8.1 Co_Occurrence matrix

From Figure **(3.8.1.1)** First 10 words are visualized from positive co_occurrence matrix.In Figure **(3.8.1.1)** Each words taken as row and column from pre processed positive dataset.The indices value of *words*[*i*] and *words*[*j*] is co_occurrence of both words.

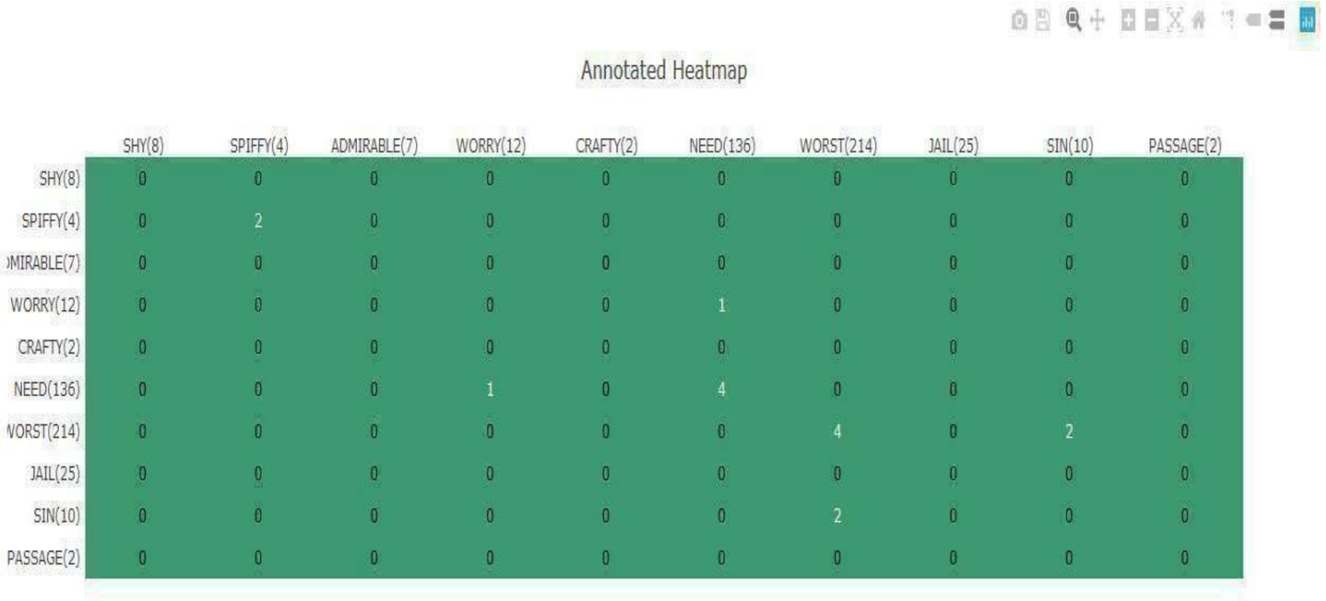


Annotated Heatmap

	HAIRY(2)	WORSE(59)	SHY(20)	WORRY(15)	ADMIRABLE(13)	CRAFTY(2)	STRIDE(6)	NEED(119)	WORST(43)	JAIL(35)
HAIRY(2)	0	0	0	0	0	0	0	0	0	0
WORSE(59)	0	4	0	0	0	0	0	0	0	0
SHY(20)	0	0	0	0	0	0	0	0	0	0
WORRY(15)	0	0	0	0	0	0	0	1	0	0
IIRABLE(13)	0	0	0	0	0	0	0	0	0	0
CRAFTY(2)	0	0	0	0	0	0	0	0	0	0
STRIDE(6)	0	0	0	0	0	0	0	0	0	0
NEED(119)	0	0	0	1	0	0	0	0	0	0
WORST(43)	0	0	0	0	0	0	0	0	0	0
JAIL(35)	0	0	0	0	0	0	0	0	0	0

Fig 3.8.1.1 : Sample View of positive co_occurrence Matrix With First 10 Tokens.

From Figure **(3.8.1.2)** First 10 words are visualized from negative co_occurrence matrix.In Figure **(3.8.1.2)** Each words taken as row and column from pre processed negative dataset.The indices value of *words*[*i*] and *words*[*j*] is co_occurrence of both words.



Annotated Heatmap

	SHY(8)	SPIFFY(4)	ADMIRABLE(7)	WORRY(12)	CRAFTY(2)	NEED(136)	WORST(214)	JAIL(25)	SIN(10)	PASSAGE(2)
SHY(8)	0	0	0	0	0	0	0	0	0	0
SPIFFY(4)	0	2	0	0	0	0	0	0	0	0
)MIRABLE(7)	0	0	0	0	0	0	0	0	0	0
WORRY(12)	0	0	0	0	0	1	0	0	0	0
CRAFTY(2)	0	0	0	0	0	0	0	0	0	0
NEED(136)	0	0	0	1	0	4	0	0	0	0
VORST(214)	0	0	0	0	0	0	4	0	2	0
JAIL(25)	0	0	0	0	0	0	0	0	0	0
SIN(10)	0	0	0	0	0	0	2	0	0	0
PASSAGE(2)	0	0	0	0	0	0	0	0	0	0

Fig 3.8.1.2 : Sample View of negative co_occurrence Matrix With First 10 Tokens.

3.8.2 Key Graph Analysis

From positive co_occurrence matrix a knowledge graph Figure **(3.8.2.1)** has created to visualize the density of words pair.The density area of words from Figure **(3.8.2.1)** indicates that these words has higher chance to being positive.

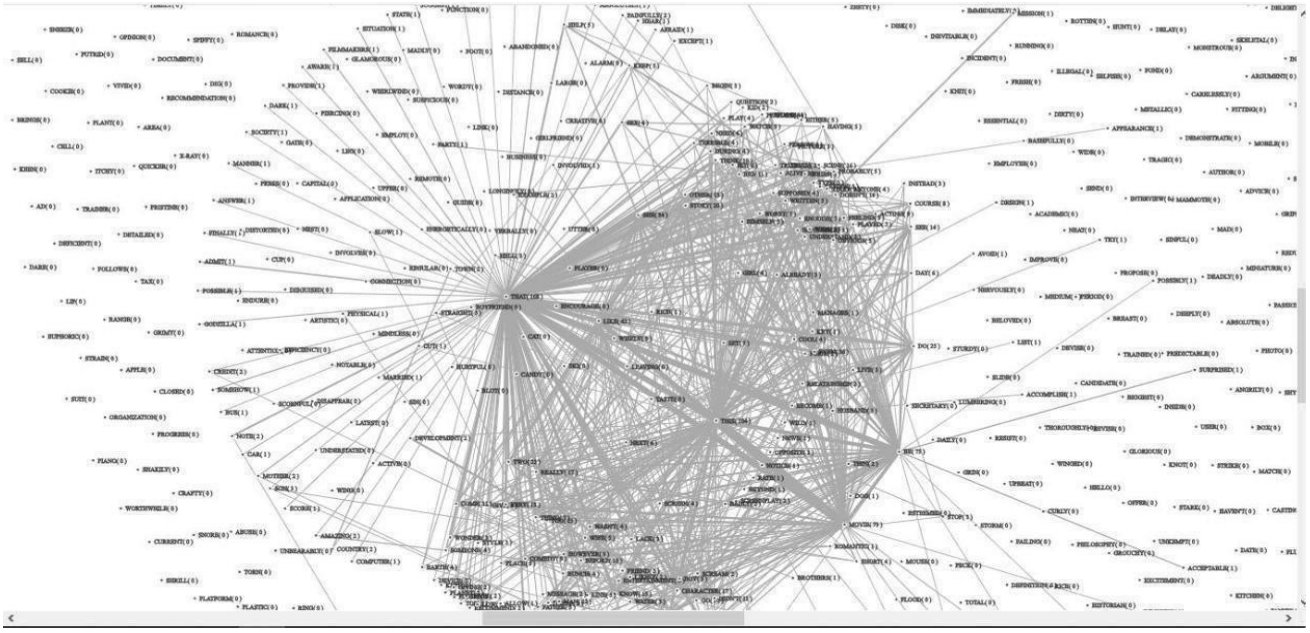


Fig 3.8.2.1 : Positive Knowledge Graph

From negative co_occurrence matrix a knowledge graph Figure **(3.8.2.2)** has created to visualize the density of words pair.The density area of words from Figure **(3.8.2.2)** indicates that these words has higher chance to being negative.

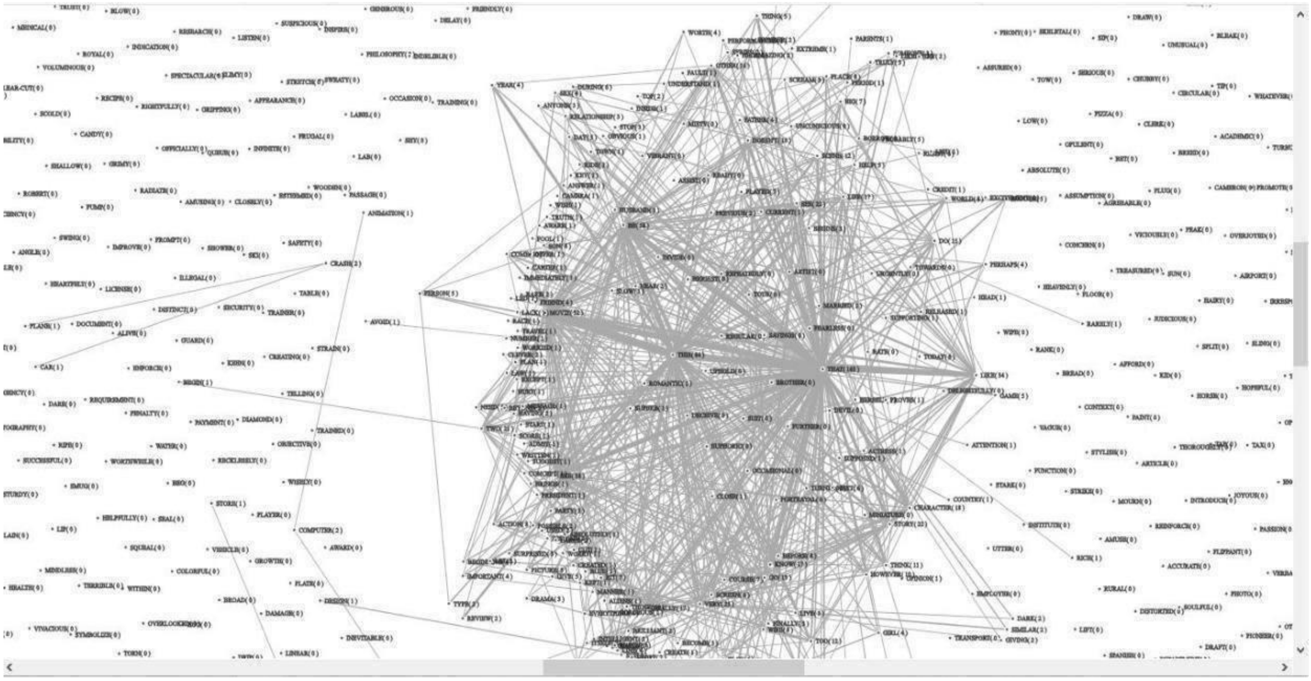


Fig 3.8.2.2: Negative Knowledge Graph

The Pair Connection “CM[i][j]” of words or tokens have been set to while visualizing minimum 5 otherwise there will be no connection between two words.The approach of Key Graph Analysis described in **section (3.6).** The subset of words those in are density of connected area are has the high chance to being positive or negative from **Fig(3.8.2.1 , 3.8.2.2).**

CHAPTER 4

RESULTS AND DISCUSSION

The Result is based on the accuracy , that a proposed model can decide the true positive and true negative result from Test Classifier.Lot of Researcher worked before on the same dataset and they have gain much comprehensive result.There are two standd movie review dataset[12] , from this dataset 80% data was trained and 20% data were tested.Fifty different set of train and test data were created by manipulating 80% train and 20% test data. From the average of all fifty set decide to a finalresult.

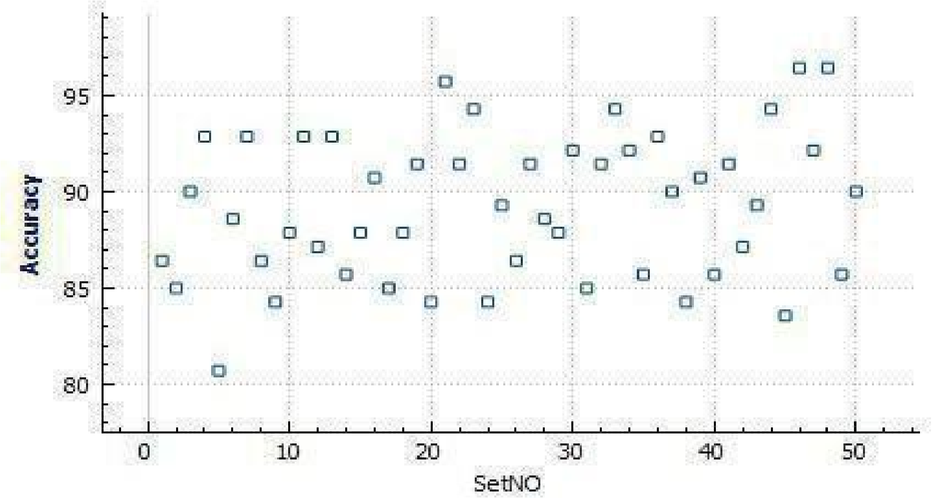
4.1 Version One Dataset Accuracy

Version One DataSet has 700 positive movie review and
700 negative movie review.

4.1.1 Positive Movie Review Accuracy

From the standard movie review version one , there were 700 positive movie review data.After Training the version one dataset (**80% positive and negative**) , the test of 50 k fold cross validation accuracy of positive set / True Positive Accuracy is givenbelow,

Figure 4.1: Fifty Different Accuracy from Version One Positive Review

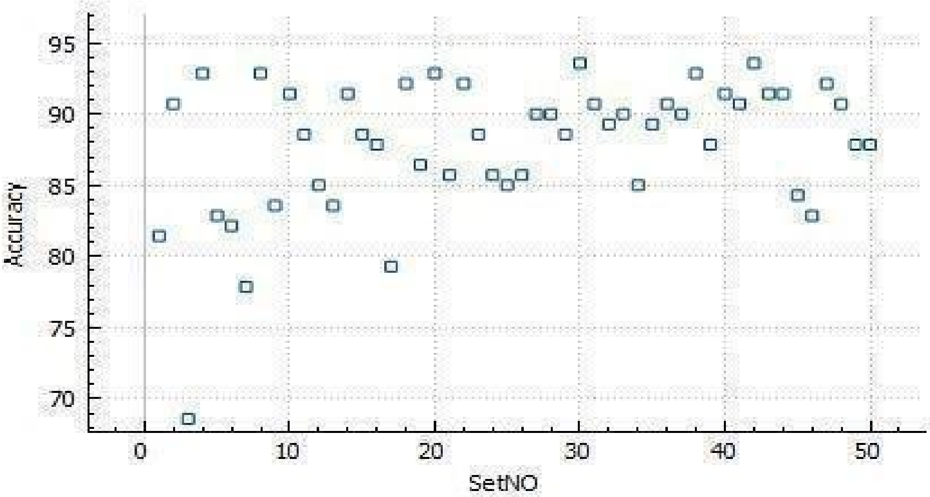


Positive Average Accuracy From Data Set Version One = 89.21428571428571

4.1.2 Negative Movie Review Accuracy

From the standard movie review version one , there were 700 negative movie review data.After Training the version one dataset (**80% positive and negative**) , the test of 50 k fold cross validation accuracy of negative set / True Negative Accuracy is given below,

Figure 4.2: Fifty Different Accuracy from Version One Negative Review



Negative Average Accuracy From Data Set Version One = 87.9

Total Average Accuracy = (True_Positive+True_Negative)/2

Total Average Accuracy= (89.2142857142857l+87.9)/2 ≅ 88.56

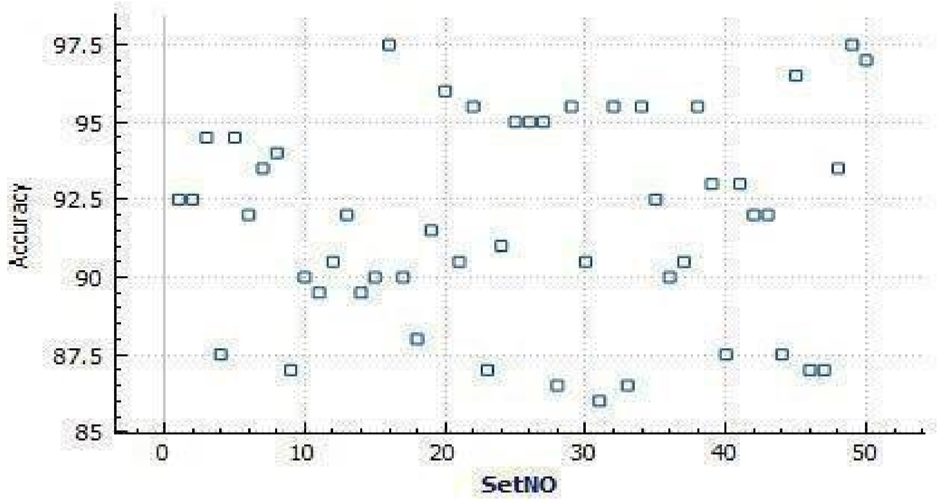
4.2 Version Two Dataset Accuracy

Version Two has 1000 positive review and 1000 negative review.

4.2.1 Positive Movie Review Accuracy

From the standard movie review version two , there were 1000 positive movie review data.After Training the version two dataset (**80% positive and negative**) , the test of 50 k fold cross validation accuracy of positive set / True Positive Accuracy is given below,

Figure 4.2: Fifty Different Accuracy from Version Two Positive Review

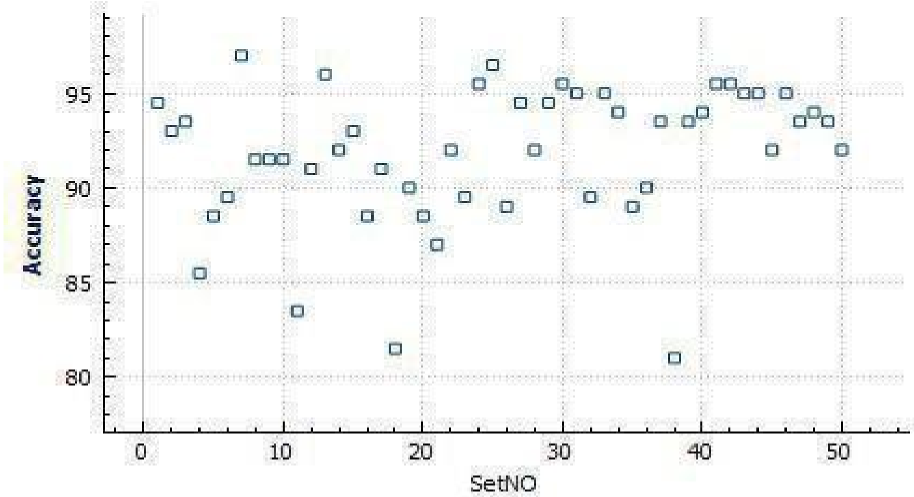


Positive Average Accuracy From Data Set Version Two = 91.79

4.2.2 Negative Movie Review Accuracy

From the standard movie review version two , there were 1000 negative movie review data.After Training the version two dataset (**80% positive and negative**) , the test of 50 k fold cross validation accuracy of negative set / True Negative Accuracy is given below,

Figure 4.4: Fifty Different Accuracy from Version Two Negative Review



Negative Average Accuracy From Data Set Version Two = 91.85

Total Average Accuracy = (True_Positive+True_Negative)/2

Total Average Accuracy = (91.79+91.85)/2 91.82

CHAPTER 5

CONCLUSIONS AND RECOMMENDATION

5.1 Conclusion

This paper focused on knowledge graph algorithm[15] and the subset of words. With these subset of words two co_occurrence matrix[18] was created to perform probabilistic math to find out the result. Its a process that refers to natural language processing[13]. Co_occurrence matrix[18] is very common in NLP[13]. Word embedding[19] , Word2Vec[16] algorithm also focuses on co_occurrence matrix[18].On the other hand jaccard distance[17] use for find out the combinational similarity of sets. Because of the fact the subset of words contained so much common words between to sets so that's why jaccard distance used for find out the dissimilarity between to sets.

5.2 Recommendation for Future Works

Previously Mentioned in Data Pre Processing the occurrence ratio of words kept"30/560" just analyzing a single sample test case. After watching first hundred iteration "*value*[*i*]/*N* ; *where* N=Total_Test_DataSet" this ratio was decided.

This ratio can be change by every different subset to get more promising result. More importantly for each different dataset this ratio needs to get changed. So in future their will be a process so that propose model can change the ratio by its own.Also in same fashion previously mentioned in jaccard distance the difference ratio of two set kept 0.005 after analyzing the sample test case by watching different iteration.

REFERENCES

1. A. Go, R.Bhayani, and L.Huang, Twitter sentiment classification using distant supervision, CS224N Project Report, Stanford, 2009.

2. B. Liu, Sentiment Analysis and Opinion Mining, Morgan Claypool Publishers, May 2012, pp. 1-167.

3. B. Pang , L. Lee, and S. Vaithyanathan, Thumbs up?:sentiment classification using machine learning techniques, Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol.10, 2002, pp. 79-86.

4. P. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews Proceedings of Annual Meeting of the Association for Computational Linguistics, 2002.

5. A. Abbasi, H Chen, A Salem, Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums, ACM Trans. Inf. Syst. 26, 3, Article 12 (June2008).

6. Struart Russell,Peter Norvig,Artificial Intelligence A Modern Approach,Education Inc 2010 , pp. 495-499

7. Sun, A. (2012). Short text classification using very few words. Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '12.

8. Islam, M. R., Kabir, M. A., Ahmed, A., Kamal, A. R. M., Wang, H., & Ulhaq, A. (2018). Depression detection from social network data using machine learning techniques. Health Information Science and Systems, 6(1).

9. T. Mullen, N. Collier, Incorporating topic information into sentiment analysis models, In Proceedings of the ACL 2004 on Interactive poster and demonstration sessions , Article25

10. E. Riloff, S. Patwardhan, J. Wiebe, Feature subsumption for opinion analysis , ACL, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 440-448, 2006.

11. Hitesh H Parmar,Sanjay Bhanderi,Glory Shah.(2014).Sentiment Mining of Movie Review using Random Forest with Tuned Hyperparameters.Conference: International Conference on Information Science At: Kerala

12. Dataset : cs.cornell.edu/people/pabo/moviereview-data/.

13. Struart Russell,Peter Norvig,Artificial Intelligence A Modern Approach,Education Inc 2010 , pp. 860-882

14. Struart Russell,Peter Norvig,Artificial Intelligence A Modern Approach,Education Inc 2010 , pp. 865,882

15. Takita, M., Tanaka, Y., Kodama, Y., Murashige, N., Hatanaka, N., Kishi, Y., … Kami, M. (2011). Data mining of mental health issues of non-bone marrow donor siblings. Journal of Clinical Bioinformatics, 1(1), 19.

16. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean.(2013).Efficient Estimation of Word Representations in Vector Space.Cornell University.

17. X. Bai, Predicting consumer sentiments from online text,
Decision Support Systems 50 (4), pp. 732-742, 2011.

18. Gotlieb, C. C., & Kreyszig, H. E. (1990). Texture descriptors based onco-occurrence matrices. Computer Vision, Graphics, and Image Processing, 51(1), 70–86.

19. Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, Bing Qin.(2014).Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification.Association for Computational Linguistics

20. https://d3js.org/d3.v2.min.js?2.9.3

21. https://cdn.plot.ly/plotly-latest.min.js

22. Zhang, C., Zeng, D., Li, J., Wang, F.-Y., & Zuo, W. (2009). Sentiment analysis of Chinese documents: From sentence to document level. Journal of the American Society for Information Science and Technology, 60(12), 2474–2487.

23. Boiy, E., & Moens, M.-F. (2008). A machine learning approach to sentiment analysis in multilingual Web texts. Information Retrieval, 12(5), 526–558.

24. McIntyre-Bhatty, Y. T. (2000). Neural network analysis and the characteristics ofmarket sentiment in the financial markets. Expert Systems, 17(4), 191–198.

25. Kang, H., Yoo, S. J., & Han, D. (2012). Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. Expert Systems with Applications, 39(5), 6000–6010.

26. Chowdhury, G. G. (2005). Natural language processing. Annual Review ofInformation Science and Technology, 37(1), 51–89.

27. Zhang, Y., Jin, R., & Zhou, Z.-H. (2010). Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics, 1(1-4),43–52.

28. Shameem, M.-U.-S., & Ferdous, R. (2009). An efficient k-means algorithm integrated with Jaccard distance measure for document clustering. 2009 First Asian Himalayas International Conference on Internet.

29. Ivchenko, G. I., & Honov, S. A. (1998). On the jaccard similarity test. Journalof Mathematical Sciences, 88(6), 789–794.

30. Liu, S., Fan, X., & Chai, J. (2017). A clustering analysis of news text based on co-occurrence matrix. 2017 3rd IEEE International Conference on Computer and Communications(ICCC).

31. Liu, S., Fan, X., & Chai, J. (2017). Clustering analysis of feature words in news text based on co-occurrence matrix. 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI).

32. Ahmed, F., & Nürnberger, A. (2009). Evaluation of n-gram conflation approaches for Arabic text retrieval. Journal of the American Society for Information Science and Technology, 60(7), 1448–1465.

33. B. Pang, L. Lee, A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, Proceedings of ACL, 2004.

34. A. Kennedy, D. Inkpen, Sentiment Classification of Movie Reviews Using Contextual Valence Shifters, Computational Intelligence, Vol. 22, No. 2 . pp. 110-125. 2006.

35. Z. Jian, X. Chen, Han-shi, Sentiment classification using the theory of ANNs, The Journal of China Universities of Posts and Telecommunications, 17(Suppl.): 58 - 62, 2010.

36. R. Xia, C. Zonga, S. Li, Ensemble of feature sets and classification algorithms for sentiment classification, Information Sciences, Elsevier, 181, PP.1138-1152, 2011.

37. B. Agarwal, N. Mittal, Sentiment Classification using Rough Set based Hybrid Feature Selection, WASSA 2013: 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis

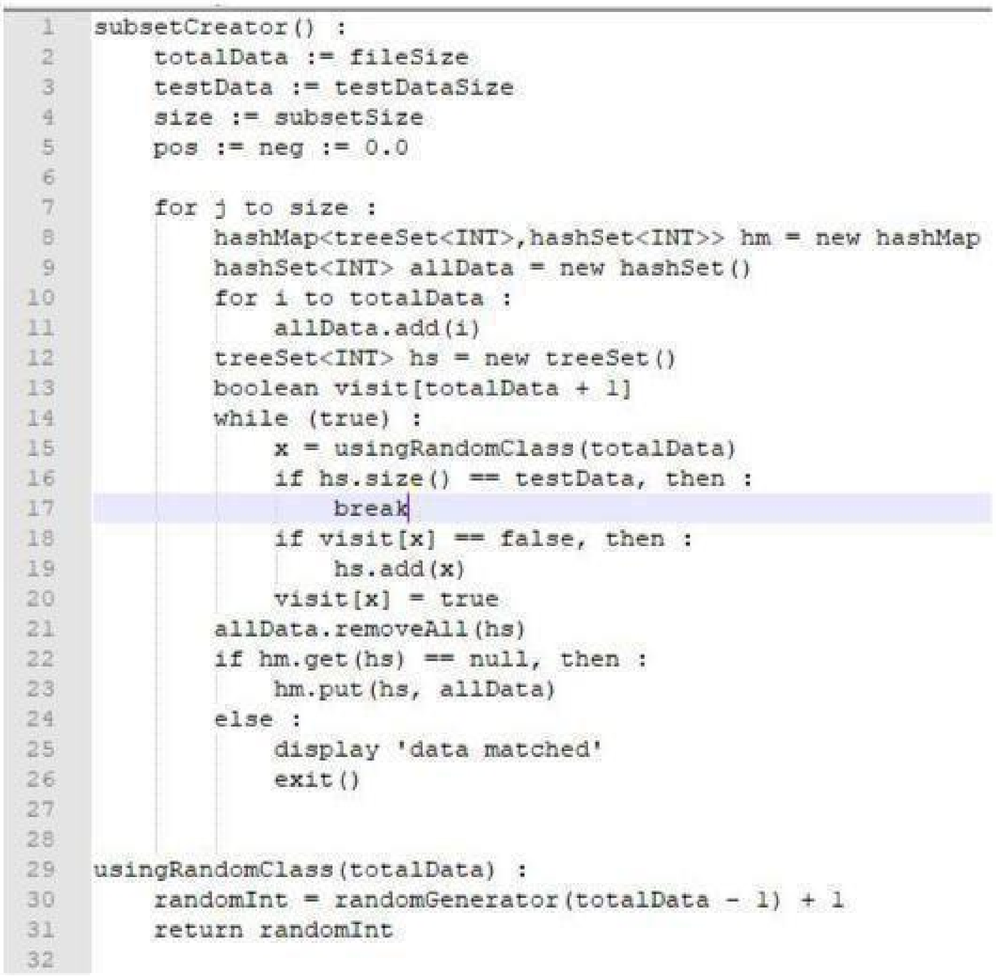
38. R. Prabowo, M. Thelwall, Sentiment analysis: A combined approach, Journal of Informatics, Volume 3, Issue 2, PP.
143-157, 2009.

39. A. Sharma, S. Dey, Comparative Study of Feature Selection and Machine Learning Techniques for Sentiment Analysis ,Proceedings of the 2012 ACM Research in Applied Computation Symposium, ACM, New York, NY, USA, 1-7.

40. A. Konig, E. Brill, Reducing the human overhead in text categorization, In Proceedings of the 12th ACM SIGKDD conference on knowledge discovery and data mining, pp. 598- 603,2006.

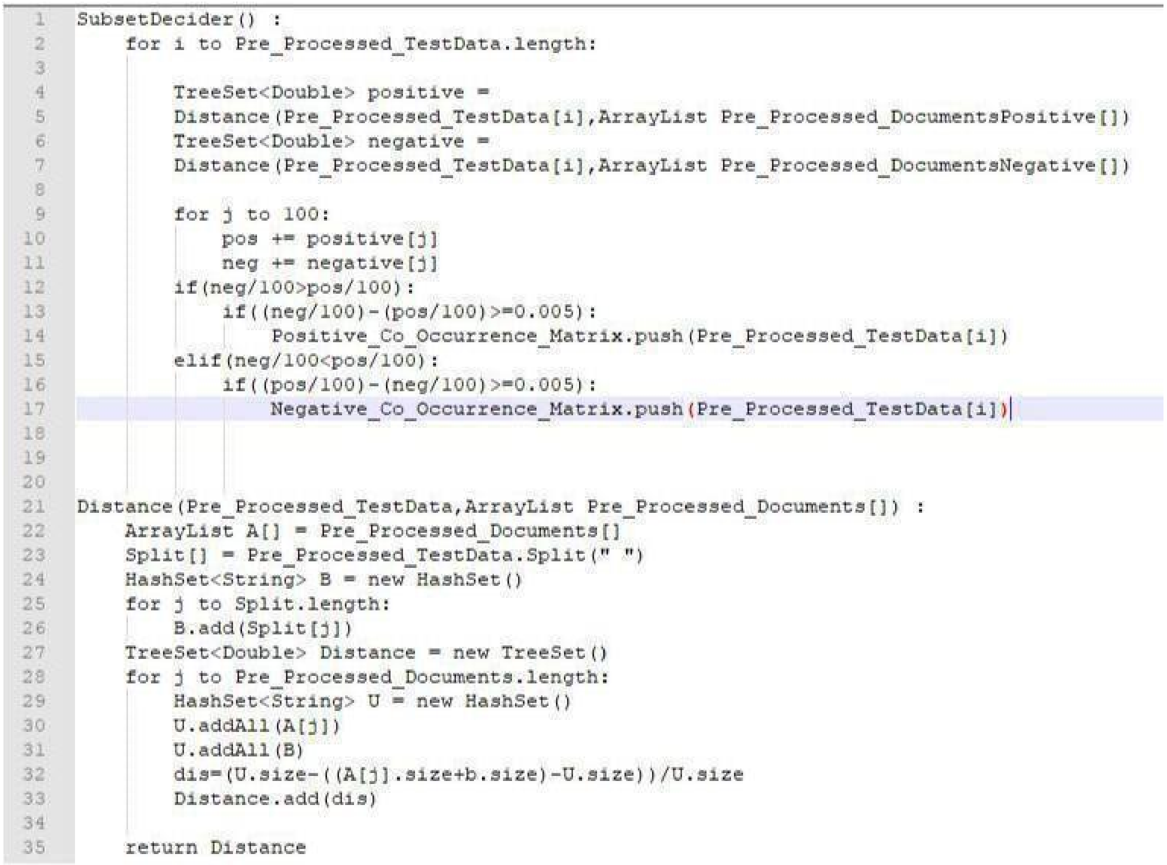
Appendix A

Algorithm 1 : Train And Test DataSet



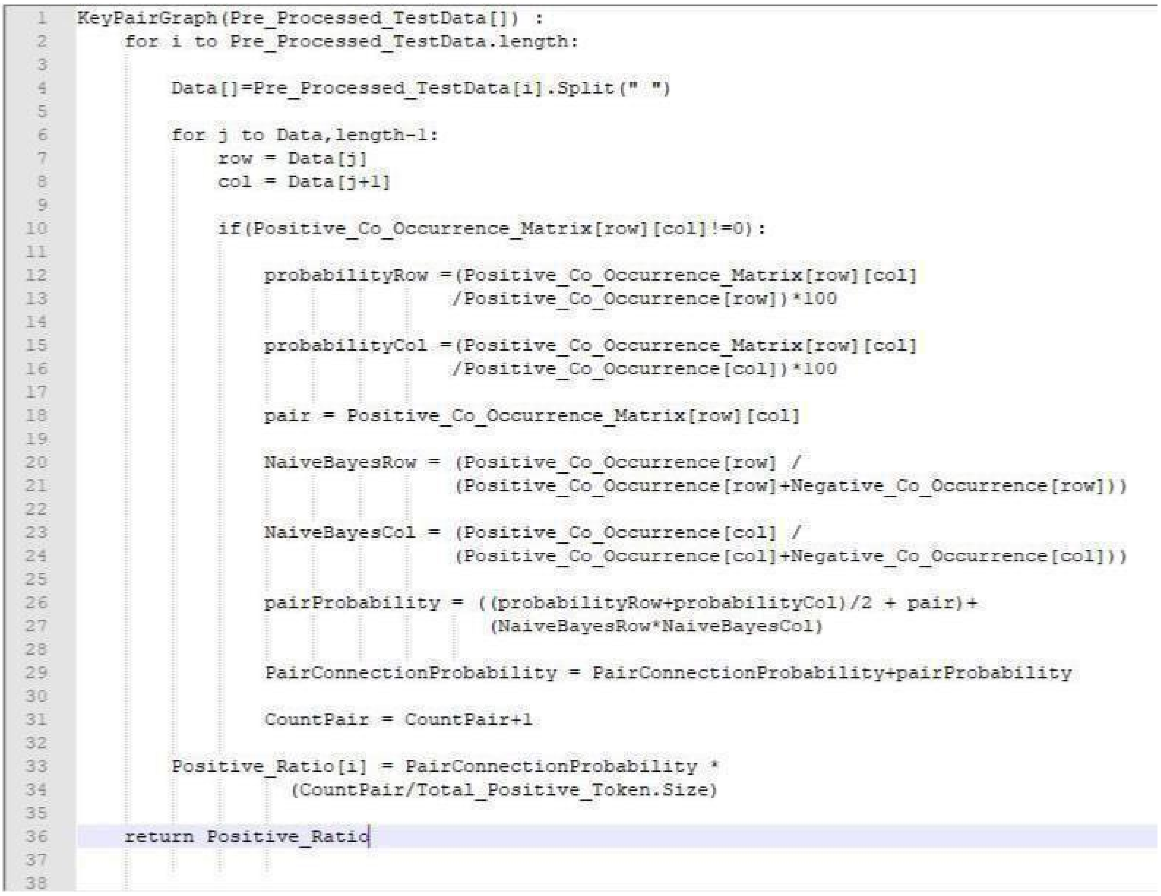
```
subsetCreator() :
    totalData := fileSize
    testData := testDataSize
    size := subsetSize
    pos := neg := 0.0

    for j to size :
        hashMap<treeSet<INT>,hashSet<INT>> hm = new hashMap
        hashSet<INT> allData = new hashSet()
        for i to totalData :
            allData.add(i)
        treeSet<INT> hs = new treeSet()
        boolean visit[totalData + 1]
        while (true) :
            x = usingRandomClass(totalData)
            if hs.size() == testData, then :
                break
            if visit[x] == false, then :
                hs.add(x)
            visit[x] = true
        allData.removeAll(hs)
        if hm.get(hs) == null, then :
            hm.put(hs, allData)
        else :
            display 'data matched'
            exit()


usingRandomClass(totalData) :
    randomInt = randomGenerator(totalData - 1) + 1
    return randomInt

```

Algorithm 2: Jaccard Distance



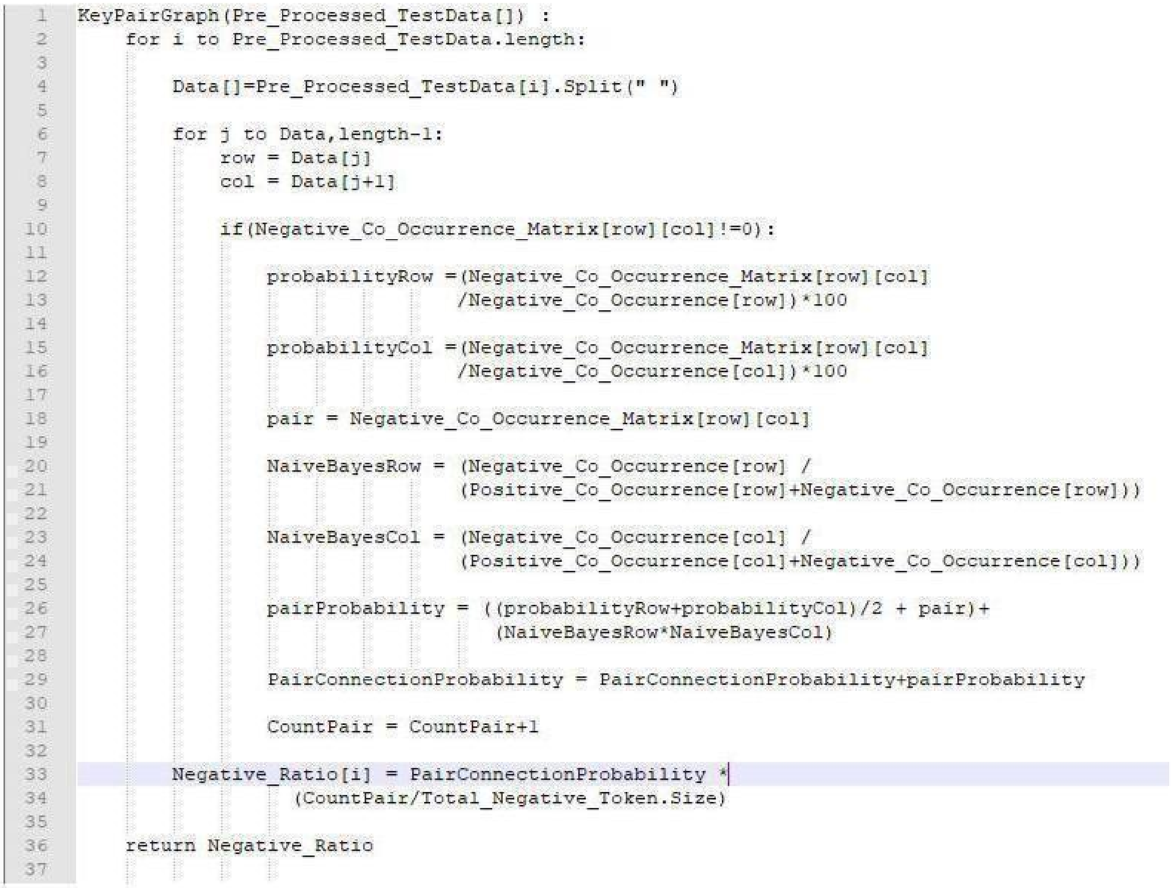
```
SubsetDecider() :
    for i to Pre_Processed_TestData.length:

        TreeSet<Double> positive =
        Distance(Pre_Processed_TestData[i],ArrayList Pre_Processed_DocumentsPositive[])
        TreeSet<Double> negative =
        Distance(Pre_Processed_TestData[i],ArrayList Pre_Processed_DocumentsNegative[])

        for j to 100:
            pos += positive[j]
            neg += negative[j]
        if(neg/100>pos/100):
            if((neg/100)-(pos/100)>=0.005):
                Positive_Co_Occurrence_Matrix.push(Pre_Processed_TestData[i])
        elif(neg/100<pos/100):
            if((pos/100)-(neg/100)>=0.005):
                Negative_Co_Occurrence_Matrix.push(Pre_Processed_TestData[i])



Distance(Pre_Processed_TestData,ArrayList Pre_Processed_Documents[]) :
    ArrayList A[] = Pre_Processed_Documents[]
    Split[] = Pre_Processed_TestData.Split(" ")
    HashSet<String> B = new HashSet()
    for j to Split.length:
        B.add(Split[j])
    TreeSet<Double> Distance = new TreeSet()
    for j to Pre_Processed_Documents.length:
        HashSet<String> U = new HashSet()
        U.addAll(A[j])
        U.addAll(B)
        dis=(U.size-((A[j].size+b.size)-U.size))/U.size
        Distance.add(dis)

    return Distance
```

Algorithm 3: Calculating Positive Key Pair Ratio



```
KeyPairGraph(Pre_Processed_TestData[]) :
    for i to Pre_Processed_TestData.length:

        Data[]=Pre_Processed_TestData[i].Split(" ")

        for j to Data,length-1:
            row = Data[j]
            col = Data[j+1]

            if(Positive_Co_Occurrence_Matrix[row][col]!=0):

                probabilityRow =(Positive_Co_Occurrence_Matrix[row][col]
                                /Positive_Co_Occurrence[row])*100

                probabilityCol =(Positive_Co_Occurrence_Matrix[row][col]
                                /Positive_Co_Occurrence[col])*100

                pair = Positive_Co_Occurrence_Matrix[row][col]

                NaiveBayesRow = (Positive_Co_Occurrence[row] /
                                (Positive_Co_Occurrence[row]+Negative_Co_Occurrence[row]))

                NaiveBayesCol = (Positive_Co_Occurrence[col] /
                                (Positive_Co_Occurrence[col]+Negative_Co_Occurrence[col]))

                pairProbability = ((probabilityRow+probabilityCol)/2 + pair)+
                                   (NaiveBayesRow*NaiveBayesCol)

                PairConnectionProbability = PairConnectionProbability+pairProbability

                CountPair = CountPair+1

        Positive_Ratio[i] = PairConnectionProbability *
                  (CountPair/Total_Positive_Token.Size)

    return Positive_Ratio
```

Algorithm 4: Calculating Negative Key Pair Ratio



```
KeyPairGraph(Pre_Processed_TestData[]) :
    for i to Pre_Processed_TestData.length:

        Data[]=Pre_Processed_TestData[i].Split(" ")

        for j to Data,length-1:
            row = Data[j]
            col = Data[j+1]

            if(Negative_Co_Occurrence_Matrix[row][col]!=0):

                probabilityRow =(Negative_Co_Occurrence_Matrix[row][col]
                                /Negative_Co_Occurrence[row])*100

                probabilityCol =(Negative_Co_Occurrence_Matrix[row][col]
                                /Negative_Co_Occurrence[col])*100

                pair = Negative_Co_Occurrence_Matrix[row][col]

                NaiveBayesRow = (Negative_Co_Occurrence[row] /
                                (Positive_Co_Occurrence[row]+Negative_Co_Occurrence[row]))

                NaiveBayesCol = (Negative_Co_Occurrence[col] /
                                (Positive_Co_Occurrence[col]+Negative_Co_Occurrence[col]))

                pairProbability = ((probabilityRow+probabilityCol)/2 + pair)+
                                   (NaiveBayesRow*NaiveBayesCol)

                PairConnectionProbability = PairConnectionProbability+pairProbability

                CountPair = CountPair+1

        Negative_Ratio[i] = PairConnectionProbability *
                  (CountPair/Total_Negative_Token.Size)

    return Negative_Ratio
```