

A MAJOR PROJECT REPORT
on
**SENTIMENT CLASSIFICATION
OF HOTEL REVIEWS**

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology
In
Computer Science Engineering

Under The Guidance Of
MRS. POONAM VERMA

Submitted By

SAKSHAM ARORA
(02351202713)

VINIT GOEL
(02851202713)

ANKIT JAIN
(03051202713)

SHASHANK SHEKHAR
(04151202713)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING

A-4, PASCHIM VIHAR

DELHI

April, 2017

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this report entitled "**SENTIMENT CLASSIFICATION OF HOTEL REVIEWS**", in fulfillment of the requirement for the award of the degree Bachelor of Technology in Computer Science & Engineering, submitted in CSE Department, BVCOE affiliated to Guru Gobind Singh Indraprastha University, New Delhi, is an authentic record of our own work carried out during my degree under the guidance of **Mrs. Poonam Verma.**

The work reported in this has not been submitted by us for award of any other degree or diploma.

Date : VINIT GOEL

Place: (02851202713)

ANKIT JAIN

(03051202713)

SAKSHAM ARORA

(02351202713)

SHASHANK SHEKHAR

(04151202713)

CERTIFICATE

This is to certify that the project work entitled "**SENTIMENT CLASSIFICATION OF HOTEL REVIEWS**" submitted by **Vinit Goel(028)** ,**Ankit Jain(030)** , **Shashank Shekhar(041)** & **Saksham Arora(023)** in fulfilment for the requirements of the award of Bachelor of Technology Degree in Computer Science & Engineering at BVCOE, New Delhi is an authentic work carried out by them under my supervision and guidance. To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any degree .

DATE:

Mrs. Poonam Verma

(Project Guide)

Mrs. Narina Thakur

(HOD,CSE)

ACKNOWLEDGEMENT

We express our sincere gratitude to Mrs. Narina Thakur (HOD, CSE) and Mrs. Poonam Verma (Asst. Prof. ,CSE), for their valuable guidance and timely suggestions during the entire duration of our project work, without which this work would not have been possible. We would also like to convey our deep regards to all other faculty members of CSE department, who have bestowed their great effort and guidance at appropriate times without which it would have been very difficult on our part to finish this work. Finally we would also like to thank our friends for their advice and pointing out our mistakes.

VINIT GOEL

(02851202713)

ANKIT JAIN

(03051202713)

SAKSHAM ARORA

(02351202713)

SHASHANK SHEKHAR

(04151202713)

ABSTRACT

Sentiment analysis refers to a reckoning process that deals with treatment of opinions, emotions, and subjectivity . We contemplate problem of classifying a hotel review into sentimental categories like positive , neutral, and negative and thereby extracting sentiment of consumer and predicting behaviour on four traits : food, service, price, and locality that defines a hotel . Using Hotel reviews Data set from Trip Advisor and SentiWordnet dictionary (an opinion lexicon derived from WordNet database), our approach's sole focus is to gauge consumer sentiment and forecasting comportment using both lexicon features and machine learning algorithms . We conclude by devising a hybrid approach that collectively exhibit accuracy of machine learning techniques and speed of a lexicon approach .

TABLE OF CONTENTS

<i>Title</i>	i
<i>Declaration</i>	ii
<i>Certificate</i>	iii
<i>Acknowledgement</i>	iv
<i>Abstract</i>	v
<i>Table of Content</i>	vi
<i>List of Images</i>	viii
1. INTRODUCTION	1
2. LITERATURE SURVEY	2
2.1 Java	2
2.2 R(Language & Studio)	2
3. Sentiment Analysis Technique	7
3.1 Lexical Analysis	7
3.2 Machine Learning Based Techniques	13
3.2.1 Naïve Bayesian	14
3.2.2 SVM	15
3.2.3 Random Forest	16
3.3 Hybrid Analysis	22
4. METHODOLOGY	24
4.1 Proposed System Architecture	24
4.2 The Dataset	26
4.3 Manually Annotated Dataset	27
4.4 Developing the Classifier	29
5. OUTPUT SCREEN	42
6. CONCLUSION	47
7. REFERENCES & BIBLIOGRAPHY	48

LIST OF IMAGES

Fig. 3.1 Working of a lexical technique	7
Fig. 3.2 shows the typical number of steps involved in a machine learning technique.....	13
Fig 3.3 SVM.....	16
Fig. 3.4 Hybrid Technique.....	22
Fig. 4.1 One of the files of the dataset used.....	26
Fig. 4.2 One of the review files after manual annotation.....	27
Fig. 4.3 Review with tags.....	28
Fig. 4.4 Working of the classifier.....	29
Fig. 4.5 A section of the code that was used to clean SentiWordNet.....	30
Fig. 4.6 SentiWordNet (as it looked before cleaning).....	31
Fig. 4.7 Cleaned SentiWordNet.....	31
Fig.4.8 Trie Node Class.....	32
Fig.4.9 Create Trie method of the Classifier.....	33
Fig.4.10 Classify method of the Classifier.....	33
Fig.4.11 Classified File (the file containing classifications).....	34
Fig.4.12 Calculating SWN efficiency.....	34
Fig.4.13 MANUALLY ANNOTATED DATASET.....	35
Fig.4.14 CLEANING DATASET.....	35
Fig.4.15 CLEANED DATASET.....	36
Fig.4.16 CREATING CSV FOR TRAINING MODEL.....	36
Fig.4.17 CSV FILE OF DATASET.....	37
Fig. 4.18 ADDING AVERAGE POSITIVE & NEGATIVE SCORES IN DATASET.....	37
Fig.4.19 HYBRID DATASET.....	38
Fig.4.20 EFFICIENCY OF LEXICON CLASSIFIER.....	40
Fig.4.21 RECALL ACCURACY.....	40
Fig. 4.22NAÏVE BAYES WITHOUT NEUTRAL CLASS.....	41
Fig.5.1 COMPOSITION OF DATASET.....	43
Fig.5.2 PLOT OF NAÏVE BAYES MACHINE LEARNING TECHNIQUE.....	44
Fig.5.3PLOT OF NAÏVE BAYES HYBRID TECHNIQUE.....	44
Fig. 5.4 COMPARING RECALL ACCURACIES OF NAÏVE BAYES , SVM & RANDOM FOREST WITH THEIR HYBRID ACCURACIES.....	45
Fig 5.5 COMPARING RECALL ACCURACIES OF NAÏVE BAYES WITHOUT NUETRAL CLASS.....	46

1. INTRODUCTION

With a huge amount of data available about the hotel reviews, proper classification to indicate the exact entity that causes the guest to be happy or that could be problematic can be done. Opinion Mining is a phenomenon to guess or predict the meaning of what the customer is trying to imply through its comments. A hotel consists of a lot of amenities for a guest. All these amenities are responsible for deciding that whether a guest would like it there or not. The facilities & regards the guest get, forms an impression in the heart & brain either positive or negative. The Classifier aims at locating the exact root of guest's happiness or in-satisfaction. It also can be used as an application in recommendation systems as the guest's verdict can help to tell the improvements that need to be done and also with its experience, other people can enjoy the amenities to the fullest. The verdict of a particular person may or may not be genuine but a whole lot can help to decide whether the hotel is the place to stay or not! This is exactly what the classifier aims at predicting . The classifier can help in reading for the valuable customers as well as it helps the hotel so as to improve and increase the pace in the cut-throat business.

There are two main approaches to Sentiment Analysis or opinion mining : machine learning based and lexicon based. Machine learning based approach deals with data collection, training data and classification while lexicon based method uses sentiment dictionary having sentiment words and correlate them with dataset to find the sentiment direction. The objective of this paper is to find out the concepts of lexical based and machine learning approaches in the field of sentiment analysis and combining them to devise a new hybrid technique that collectively provides the positive aspects of both approaches .

2. LITERATURE SURVEY

2.1 Java

Java has become enormously popular. Java is a full-featured, general-purpose programming language that is capable of developing robust mission-critical applications. Today, it is used not only for web programming, but also for developing standalone applications across platforms on servers, desktops, and mobile devices. It was used to develop the code to communicate with and control the robotic rover that rolled on Mars.

2.2 R (Language and Studio):

R is an open source programming language developed and used for Statistical computing and data analysis. As mentioned, R is freely available under GNU General Public License and though it has a Command Line Interface, many graphical interfaces and IDEs are also available. R and its libraries implement a wide variety of statistical and graphical techniques, including linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, and others.

It is easily extensible through functions and extensions, and many of R's standard functions are written in R itself, which makes it easy for users to follow the algorithmic choices made. For computationally intensive tasks, C, C++, and Fortran code can be linked and called at run time. R is highly extensible through the use of user-submitted packages for specific functions or specific areas of study. Due to its S heritage, R has stronger object-oriented programming facilities than most statistical computing languages. Extending R is also eased by its lexical scoping rules.

Also, a primary part of our project (Lexicon based sentiment analysis and consequently Hybrid sentiment analysis methods) involves using the trie data structure for fast data retrieval which

can be implemented efficiently in R thus making R language as our preferred choice for programming.

RStudio is a free and open-source integrated development environment (IDE) for R, a programming language for statistical computing and graphics. RStudio was founded by JJ Allaire, creator of the programming language ColdFusion. RStudio makes using R a lot easier, and lets us use lots of packages and extensions very easily.

We have used RStudio as the IDE for training, cross validation, text document matrix and overall hybrid sentiment analysis.

R Libraries and Packages Used:

Caret:

The caret package is a set of tools for building machine learning models in R. The name “caret” stands for Classification And REgression Training. As the name implies, the caret package gives you a toolkit for building classification models and regression models. The package utilizes a number of R packages but tries not to load them all at package start-up instead, it loads packages as needed and assumes that they are all installed.

R has many machine learning tools, but they can be extremely clumsy to work with. Caret solves this problem. To simplify the process, caret provides tools for almost every part of the model building process, and moreover, provides a common interface to these different machine learning methods.

Moreover, caret provides you with essential tools for:

- Data preparation, including: imputation, centering/scaling data, removing correlated predictors, reducing skewness
- Data splitting
- Model evaluation

– Variable selection

The train() function:

The core of caret's functionality is the train() function.

The primary feature of Caret used in the Project is the train function which can be used to "train" the Model where every step of the process can be customised by the user (e.g. resampling technique, choosing the optimal parameters etc). That is, train is the function that will "learn" the relationship between the users and their ratings. Also, Caret's createDataPartition by default does a stratified random split, thus removing any irregularities in calculations and analysis.

TM Package(Text Mining):

The tm package offers functionality for managing text documents, abstracts the process of document manipulation and eases the usage of heterogeneous text formats in R. The package has integrated database back-end support to minimize memory demands. An advanced meta data management is implemented for collections of text documents to alleviate the usage of large and with meta data enriched document sets. The package provides native support for reading in several classic file formats and there is also a plug-in mechanism to handle additional file formats.

The data structures and algorithms can be extended to fit custom demands, since the package is designed in a modular way to enable easy integration of new file formats, readers, transformations and filter operations.

TM provides easy access to preprocessing and manipulation mechanisms such as whitespace removal, stemming, or stopword deletion. Further a generic filter architecture is available in order to filter documents for certain criteria, or perform full text search. The package supports the export from document collections to term-document matrices.

RTextTools:

RTextTools is a machine learning package for automatic text classification that makes it simple for novice users to get started with machine learning, while allowing experienced users to easily experiment with different settings and algorithm combinations. The package includes nine algorithms for ensemble classification (svm, slda, boosting, bagging, random forests, glmnet, decision trees, neural networks, maximum entropy), comprehensive analytics, and thorough documentation.

The primary use of RTextTools Package is to create the text document matrix that maps the frequency of most occurring words and maps it to the users thereby helping the training model to relate the words with user ratings.

e1071:

The e1071 package contains functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier. Since, we use both SVM and Naive Bayes method in our analysis, the use of e1071 becomes essential to the program. The e1071 acts as a package that extends the features of R for the Naive Bayes classifiers.

2.3 Sentiment Classification

Sentiment refers to a thought, opinion, or idea based on a feeling about a situation, or a way of thinking about something. Also, it can be defined as the classification of an object regardless of whether the sentence at the facility includes a sentence in a positive feedback or negative feedback can be on it. Sentiments can be analysed and can be used for many purposes. It is traditionally used for automatic extraction of opinions types about a product and for representing positive or negative features of a product. It is widely believed that Sentiment

analysis is needed and useful. It is also broadly agreed that extracting sentiment from text is a hard semantic problem even for human beings. The concept of opinion mining is nothing but web data mining. The results of opinion mining can be utilized in various ways such as to improve the particular features by using a summary of the opinion on features.

Sentiment classification is a task of opinion mining whose objective is to classify the text in a given document to determine the overall sentiment direction. Various opinions in the text are assumed to be subjective in nature and can be categorized on a judgment factor (positive or negative) and the degree or strength to which a word or sentence in a document is positive or negative. It has been practiced on various fields such as movie reviews, hotel reviews, product reviews and services.

3. SENTIMENT ANALYSIS TECHNIQUES

3.1 LEXICAL ANALYSIS

This technique is governed by the use of a dictionary consisting pre-tagged lexicons. The input text is converted to tokens by the Tokenizer. Every new token encountered is then matched for the lexicon in the dictionary. If there is a positive match, the score is added to the total pool of score for the input text. For instance if “dramatic” is a positive match in the dictionary then the total score of the text is incremented. Otherwise the score is decremented or the word is tagged as negative. Though this technique appears to be amateur in nature, its variants have proved to be worthy.

Following Fig.3.1 shows the working of a lexical technique.

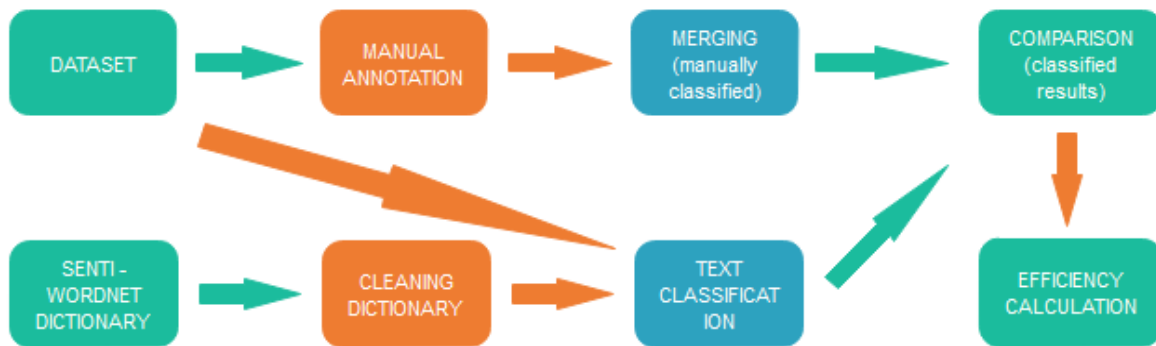


Fig. 3.1 Working of a lexical technique

The classification of a text depends on the total score it achieves. Lexical analysis has a **limitation**: its performance (in terms of time complexity and accuracy) degrades drastically with the exponential growth of the size of dictionary (number of words).

Opinions & Opinion Lexicon

Opinion as the dictionary dictates is a view or judgment formed about something, not compulsorily based on knowledge or expert advice on an official matter or facts. Similarly, A lexicon is the words used by a person, language, or vocabulary used in various fields. In grammar, a lexicon is a language's catalogue of lexemes. In any sentence, there exist certain words that represent or throw some light on objectification features of some objects written or given by any person. It may be said as a person's point of view. It can have any value such as positive, negative or neutral. Many sentiment applications rely on lexicons to supply features to a model. Publicly available resources and their relationships can be reviewed by some available ways and it looks to identify some best approaches for using sentiment lexicons in a more productive manner.

Manually built lexicons generally contain less number of terms and building manually is generally prejudiced to annotator and also a time consuming task. To get rid of these problems, lexical analysis approaches are proposed. Opinion lexicons are the combination of the words and sentiment orientation that are used in opinion mining research works. Lexical analysis approaches has the vision of extending the limited data set containing only seed terms to large size opinion lexicons. Various different works and approaches had been seen in this area.

The type of opinions that we have discussed so far is called regular opinion (Liu, 2006 and 2011). Another type is called comparative opinion (Jindal and Liu, 2006b).

Opinion in the literature and it has **two main sub-types** (Liu, 2006 and 2011):

Direct opinion: A direct opinion refers to an opinion expressed directly on an entity or an entity aspect, e.g., “The picture quality is great.”

Indirect opinion: An indirect opinion is an opinion that is expressed indirectly on an entity or aspect of an entity based on its effects on some other entities. This sub-type often occurs in the medical domain. For example, the sentence “After injection of the drug, my joints felt worse”

describes an undesirable effect of the drug on “my joints”, which indirectly gives a negative opinion or sentiment to the drug. In the case, the entity is the drug and the aspect is the effect on joints.

Much of the current research focuses on direct opinions. They are simpler to handle. Indirect opinions are often harder to deal with. For example, in the drug domain, one needs to know whether some desirable and undesirable state is before or after using the drug. For example, the sentence “Since my joints were painful, my doctor put me on this drug” does not express a sentiment or opinion on the drug because “painful joints” (which is negative) happened before using the drug.

Comparative opinion: A comparative opinion expresses a relation of similarities or differences between two or more entities and/or a preference of the opinion holder based on some shared aspects of the entities (Jindal and Liu, 2006a; Jindal and Liu, 2006b). For example, the sentences, “Coke tastes better than Pepsi” and “Coke tastes the best” express two comparative opinions. A comparative opinion is usually expressed using the comparative or superlative form of an adjective or adverb, although not always (e.g., prefer). Comparative opinions also have many types.

Explicit and Implicit Opinions

Explicit opinion: An explicit opinion is a subjective statement that gives a regular or comparative opinion, e.g., “Coke tastes great,” and “Coke tastes better than Pepsi.”

Implicit (or implied) opinion: An implicit opinion is an objective statement that implies a regular or comparative opinion. Such an objective statement usually expresses a desirable or undesirable fact, e.g., “I bought the mattress a week ago, and a valley has formed,” and “The battery life of Nokia phones is longer than Samsung phones.”

Explicit opinions are easier to detect and to classify than implicit opinions. Much of the current research has focused on explicit opinions. Relatively less work has been done on implicit

opinions (Zhang and Liu, 2011b). In a slightly different direction, (Greene and Resnik, 2009) studied the influence of syntactic choices on perceptions of implicit sentiment. For example, for the same story, different headlines can imply different sentiments.

Subjectivity and Emotion

There are two important concepts that are closely related to sentiment and opinion, i.e., subjectivity and emotion.

Definition (sentence subjectivity): An objective sentence presents some factual information about the world, while a subjective sentence expresses some personal feelings, views, or beliefs.

An example objective sentence is “iPhone is an Apple product.” An example subjective sentence is “I like iPhone.” Subjective expressions come in many forms, e.g., opinions, allegations, desires, beliefs, suspicions, and speculations (Riloff, Patwardhan and Wiebe,

2006; Wiebe, 2000). There is some confusion among researchers to equate subjectivity with opinionated. By opinionated, we mean that a document or sentence expresses or implies a positive or negative sentiment. The two concepts are not equivalent, although they have a large intersection. The task of determining whether a sentence is subjective or objective is called subjectivity classification (Wiebe and Riloff, 2005). Here, we should note the following; a subjective sentence may not express any sentiment. For example, “I think that he went home” is a subjective sentence, but does not express any sentiment.

Objective sentences can imply opinions or sentiments due to desirable and undesirable facts (Zhang and Liu, 2011b). For example, the following two sentences which state some facts clearly imply negative sentiments (which are implicit opinions) about their respective products because the facts are undesirable:

“The earphone broke in two days.” “I brought the mattress a week ago and a valley has formed.”

Apart from explicit opinion bearing subjective expressions, many other types of subjectivity have also been studied although not as extensive, e.g., affect, judgment, appreciation, speculation, hedge, perspective, arguing, agreement and disagreement, political stances (Alm,

2008; Ganter and Strube, 2009; Greene and Resnik, 2009; Hardisty, Boyd-Graber and Resnik, 2010; Lin et al., 2006; Medlock and Briscoe, 2007; Mukherjee and Liu.

Definition (emotion): Emotions are our subjective feelings and thoughts.

Emotions have been studied in multiple fields, e.g., psychology, philosophy, and sociology. The studies are very broad, from emotional responses of physiological reactions (e.g., heart rate changes, blood pressure, sweating and so on), facial expressions, gestures and postures to different types of subjective experiences of an individual's state of mind. Scientists have categorized people's emotions into some categories. However, there is still not a set of agreed basic emotions among researchers. Based on (Parrott, 2001), people have six primary emotions, i.e., love, joy, surprise, anger, sadness, and fear, which can be sub-divided into many secondary and tertiary emotions. Each emotion can also have different intensities.

Emotions are closely related to sentiments. The strength of a sentiment or opinion is typically linked to the intensity of certain emotions, e.g., joy and anger. Opinions that we study in sentiment analysis are mostly evaluations (although not always). According to consumer behavior research, evaluations can be broadly categorized into two types: rational evaluations and emotional evaluations (Chaudhuri, 2006).

Rational evaluation: Such evaluations are from rational reasoning, tangible beliefs, and utilitarian attitudes. For example, the following sentences express rational evaluations:

“The voice of this phone is clear,” “This car is worth the price,” and “I am happy with this car.”

Emotional evaluation: Such evaluations are from non-tangible and emotional responses to entities which go deep into people's state of mind. For example, the following sentences express emotional evaluations: “I love iPhone,” “I am so angry with their service people” and “This is the best car ever built.”

SentiWordNet

Sentiment classification concerns the use of automatic methods for predicting the orientation of subjective content on text documents, with applications on a number of areas including recommender and advertising systems, customer intelligence and information retrieval. SentiWordNet is an opinion lexicon derived from the WordNet database where each term is associated with numerical scores indicating positive and negative sentiment information.

The Lexicon approach for detecting sentiment in text present in literature concerns the use of lexical resources such as a dictionary of opinionated terms. SentiWordNet is one such resource, containing opinion information on terms extracted from the WordNet database and made publicly available for research purposes. SentiWordNet is built via a semi supervised method and could be a valuable resource for performing opinion mining tasks: it provides a readily available database of term sentiment information for the English language, and could be used as a replacement to the process of manually deriving ad-hoc opinion lexicons. In addition, SentiWordNet is built upon a semi automated process, and could easily be updated for future versions of WordNet, and for other languages where similar lexicons are available. Thus, an interesting research question is to assess how effective is SentiWordNet in the task of detecting sentiment in comparison to other methods, and what are the potential advantages that could be obtained from this approach.

In our project we employ the results of applying the SentiWordNet lexical resource to the problem of automatic sentiment classification of hotel reviews. Our approach comprises counting positive and negative term scores to determine sentiment orientation, and an improvement is presented by building a data set of relevant features using SentiWordNet as source, and applied to a machine learning classifier. We find that results obtained with SentiWordNet are in line with similar approaches using manual lexicons seen in the literature. In addition, our feature set approach yielded improvements over the baseline term counting method. The results indicate SentiWordNet could be used as an important resource for sentiment classification tasks. Additional considerations are made on possible further improvements to the method and its use in conjunction with other techniques.

3.2 Machine Learning Based Analysis

Machine learning is one of the most prominent techniques gaining interest of researchers due to its adaptability and accuracy. In sentiment analysis, mostly the supervised learning variants of this technique are employed. It comprises of three stages: Data collection, Pre-processing, Training data, Classification and plotting results. In the training data, a collection of tagged corpora is provided. The Classifier is presented a series of feature vectors from the previous data. A model is created based on the training data set which is employed over the new/unseen text for classification purpose. In machine learning technique, the key to accuracy of a classifier is the selection of appropriate features. Generally, unigrams (single word phrases), bi-grams (two consecutive phrases), tri-grams (three consecutive phrases) are selected as feature vectors. There are a variety of proposed features namely number of positive words, number of negative words, length of the document, Support Vector Machines (SVM) and Naïve Bayes (NB) algorithm.

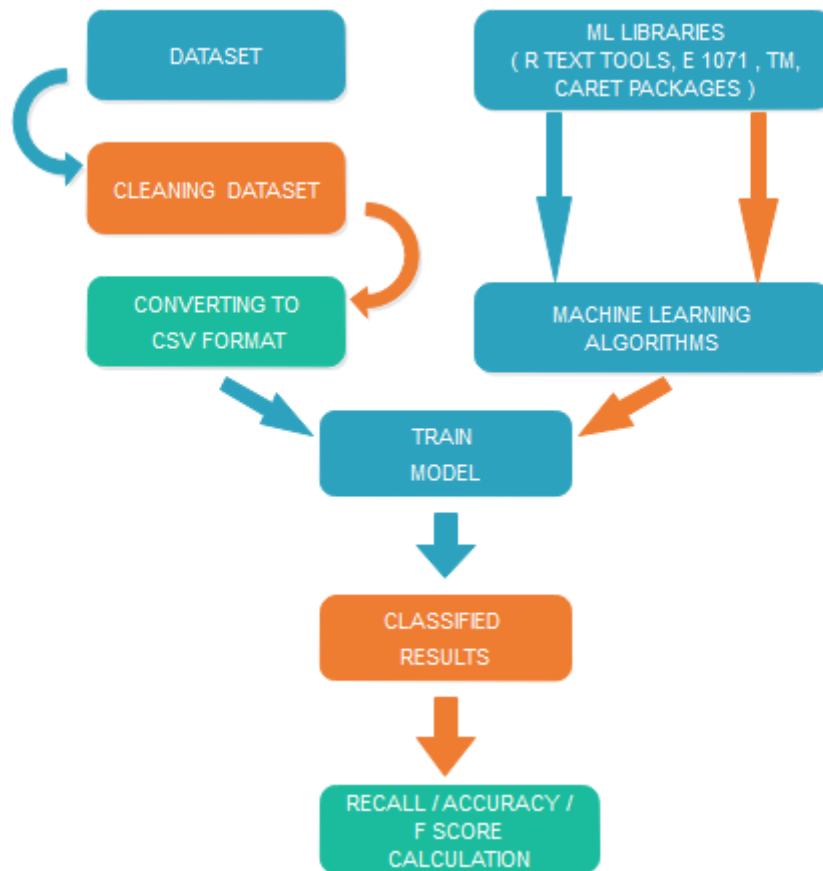


Fig. 3.2 shows the typical number of steps involved in a machine learning technique.

Working of this technique can be explained as follows:

First Step: Data Collecting – in this stage data to be analyzed is crawled from various sources like Blogs, Social networks (Twitter, MySpace, etc.) depending upon the area of application.

Second Step: Pre-processing – In this stage, the acquired data is cleaned and made ready for feeding it into the classifier. Cleaning includes extraction of keywords and symbols. For instance – Emoticons are the smiley used in textual form to represent emotions e.g. “:-)”, “:)”, “=)”, “:D”, “:-(", “:(“, “=(“, “;(", etc.. Correcting the all uppercase and all lowercase to a common case, removing the non-English (or proffered language texts), removing un-necessary white spaces and tabs, etc.

Third Step: Training Data – A hand-tagged collection of data is prepared by most commonly used crowd-sourcing method. This data is the fuel for the classifier; it will be fed to the algorithm for learning purpose.

Fourth Step: Classification – This is the heart of the whole technique. Depending upon the requirement of the application SVM or Naïve bayes is deployed for analysis. The classifier (after completing the training) is ready to be deployed to the real time tweets/text for sentiment extraction purpose.

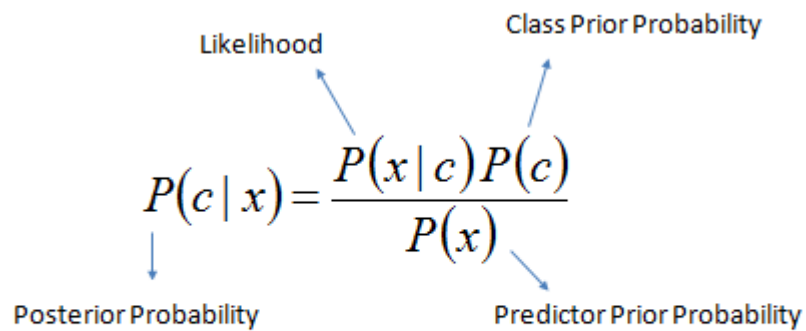
Fifth Step: Results – Results are plotted based on the type of representation selected i.e. charts, graphs, etc. Performance tuning is done prior to the release of the algorithm.

3.2.1 Naive Bayesian

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

Algorithm

Bayes theorem provides a way of calculating the posterior probability, $P(c/x)$, from $P(c)$, $P(x)$, and $P(x/c)$. Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.



The diagram shows the Bayes' Theorem formula: $P(c | x) = \frac{P(x | c)P(c)}{P(x)}$. Blue arrows point from the labels to the corresponding parts of the formula: 'Likelihood' points to $P(x | c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c | x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

- $P(c/x)$ is the posterior probability of *class (target)* given *predictor (attribute)*.
- $P(c)$ is the prior probability of *class*.
- $P(x/c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

3.2.2 Support Vector Machine

Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).

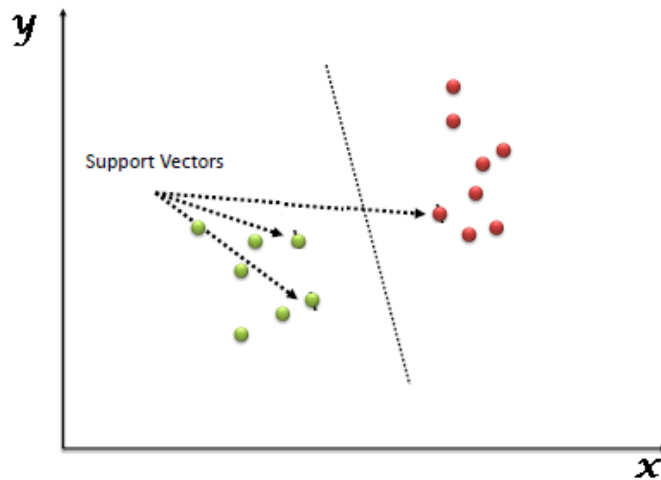


Fig 3.3 SVM

Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

3.2.3 Random Forest

They are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, and "Random Forests" is their trademark. The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho and later independently by Amit and Geman in order to construct a collection of decision trees with controlled variance.

Random Forest is considered to be a *panacea* of all data science problems. On a funny note, when you can't think of any algorithm (irrespective of situation), use random forest!

Random Forest is a versatile machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration, and does a fairly good job. It is a type of ensemble learning method, where a group of weak models combine to form a powerful model.

Term Document Matrix:

A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms. There are various schemes for determining the value that each entry in the matrix should take. One such scheme is tf-idf. They are useful in the field of natural language processing, sentiment analysis, and lexicon based algorithms.

When creating a database of terms that appear in a set of documents the document-term matrix contains rows corresponding to the documents and columns corresponding to the terms. For instance if one has the following two (short) documents:

D1 = "I like databases"

D2 = "I hate databases",

then the document-term matrix would be:

	I	like	hate	databases
D1	1	1	0	1
D2	1	0	1	1

which shows which documents contain which terms and how many times they appear.

Note that more sophisticated weights can be used; one typical example, among others, would be tf-idf.

In the current project, a similar form of term document matrix has been employed where users have been mapped against the most occurring terms in the review statement made by the user and the frequency is noted in the respective columns. This helps in training the model the relationship between the rating given by the reviewers and their choice of words in the review statement.

Document Classification:

Document classification or document categorization is a problem in library science, information science and computer science. The task is to assign a document to one or more classes or categories. This may be done "manually" (or "intellectually") or algorithmically. The intellectual classification of documents has mostly been the province of library science, while the algorithmic classification of documents is mainly in information science and computer science. The problems are overlapping, however, and there is therefore interdisciplinary research on document classification.

The documents to be classified may be texts, images, music, etc. Each kind of document possesses its special classification problems. When not otherwise specified, text classification is implied.

Documents may be classified according to their subjects or according to other attributes (such as document type, author, printing year etc.). In the rest of this article only subject classification is considered. There are two main philosophies of subject classification of documents: the content-based approach and the request-based approach. Since, the current requirement of the project is content based approach, we will discuss more on that.

Content Based Classification:

Content-based classification is classification in which the weight given to particular subjects in a document determines the class to which the document is assigned. It is, for example, a common rule for classification in libraries, that at least 20% of the content of a book should be about the class to which the book is assigned. In automatic classification it could be the number of times given words appears in a document. In our project, we employ SentiWordNet Dictionary for the classification purposes.

Performance Measures:

Confusion Matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix,[4] is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice versa).[2] The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another).

It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).

Accuracy:

Accuracy is the most direct and straightforward measure of correctness in ML. It is intuitively easy of course: we mean the proportion of correct results that a classifier achieved. If, from a

data set, a classifier could correctly guess the label of half of the examples, then we say it's accuracy was 50%. It seems obvious that the better the accuracy, the better and more useful a classifier is.

But this is not always the best measure of classifiers.

Let's delve into the possible examples cases. Either the classifier got a positive example labeled as positive, or it made a mistake and marked it as negative. Conversely, a negative example may have been (mis)labeled as positive, or correctly guessed negative. So we define the following metrics:

True Positives (TP): number of positive examples, labeled as such.

False Positives (FP): number of negative examples, labeled as positive.

True Negatives (TN): number of negative examples, labeled as such.

False Negatives (FN): number of positive examples, labeled as negative.

Precision and Recall:

$$\text{precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{recall} = \text{TP} / (\text{TP} + \text{FN})$$

Precision answers the following question: out of all the examples the classifier labeled as positive, what fraction were correct? On the other hand, recall answers: out of all the positive examples there were, what fraction did the classifier pick up?

Precisely, In pattern recognition and information retrieval binary classification, precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance.

Suppose a computer program for recognizing dogs in photographs identifies eight dogs in a picture containing 12 dogs and some cats. Of the 8 dogs identified, 5 actually are dogs (true positives), while the rest are cats (false positives). The program's precision is $5/8$ while its recall is $5/12$. When a search engine returns 30 pages only 20 of which were relevant while failing to return 40 additional relevant pages, its precision is $20/30 = 2/3$ while its recall is $20/60 = 1/3$. So, in this case, precision is "how useful the search results are", and recall is "how complete the results are".

In simple terms, high precision means that an algorithm returned substantially more relevant results than irrelevant ones, while high recall means that an algorithm returned most of the relevant results.

If the classifier does not make mistakes, then precision = recall = 1.0. But in real world tasks this is impossible to achieve. It is trivial however to have a perfect recall (simply make the classifier label all the examples as positive), but this will in turn make the classifier suffer from horrible precision and thus, turning it near useless. It is easy to increase precision (only label as positive those examples that the classifier is most certain about), but this will come with horrible recall.

The conclusion is that tweaking a classifier is a matter of balancing what is more important for us: precision or recall. It is possible to get both up: one may choose to optimize a measure that combines precision and recall into a single value, such as the F-measure.

Fscore:

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0. The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall.

$$\text{Fscore} = 2[\text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall})]$$

The F-score is widely used in machine learning. Note, however, that the F-measures do not take the true negatives into account, thus we use Fscore separately along with Precision, Recall and Accuracy. The F-score has also been widely used in the natural language processing literature, such as the evaluation of named entity recognition and word segmentation.

Cross Validation:

Cross-validation, sometimes called rotation estimation, is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (testing dataset). The goal of cross validation is to define a dataset to "test" the model in the training phase (i.e., the validation dataset), in order to limit problems like overfitting, give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem), etc.

One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

One of the main reasons for using cross-validation instead of using the conventional validation (e.g. partitioning the data set into two sets of 70% for training and 30% for test) is that there is not enough data available to partition it into separate training and test sets without losing significant modelling or testing capability. In these cases, a fair way to properly estimate model prediction performance is to use cross-validation as a powerful general technique.

In summary, cross-validation combines (averages) measures of fit (prediction error) to derive a more accurate estimate of model prediction performance

Cross-validation has the following applications:

1. Validating the robustness of a particular mining model.
2. Evaluating multiple models from a single statement.
3. Building multiple models and then identifying the best model based on statistics.

Suppose we have a model with one or more unknown parameters, and a data set to which the model can be fit (the training data set). The fitting process optimizes the model parameters to make the model fit the training data as well as possible. If we then take an independent sample of validation data from the same population as the training data, it will generally turn out that the model does not fit the validation data as well as it fits the training data. This is called overfitting, and is particularly likely to happen when the size of the training data set is small, or when the number of parameters in the model is large. Cross-validation is a way to predict the fit of a model to a hypothetical validation set when an explicit validation set is not available.

In our current project, we split the dataset into two partitions 1-800 and 801-1192 and then performs iterative cross validation to provide a robust measure that effectively demonstrates how our model will perform on an independent dataset

3.3 Hybrid Analysis

Lexicon Based Sentiment Analysis associate with the presence of certain word in document. Lexicon contains different features including the part of speech tagging of word, their sentiment values, subjectivity of word etc. The Sentiment Analysis of tweets are annotate using this features provided by these lexicons. Using that we can obtain polarity of whole tweet by averaging the sentiment values of words.

The Machine Learning based Sentiment Analysis technique requires creating a model by training the classifier with labeled examples. This means that first we require to gather a dataset with positive, negative and neutral classes, extract the features/words from that dataset & then train the algorithm based on the examples.

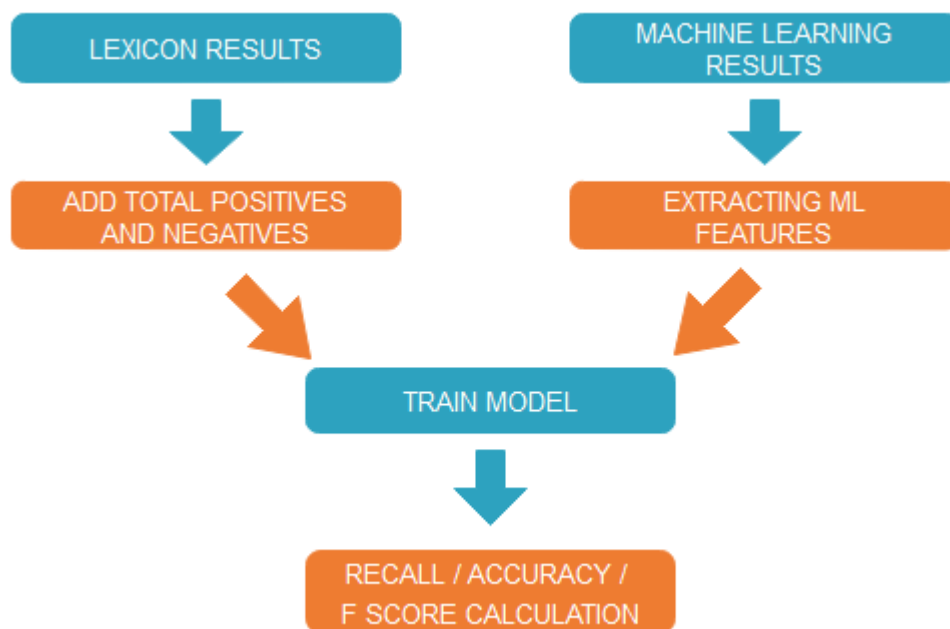


Fig. 3.4 Hybrid Technique

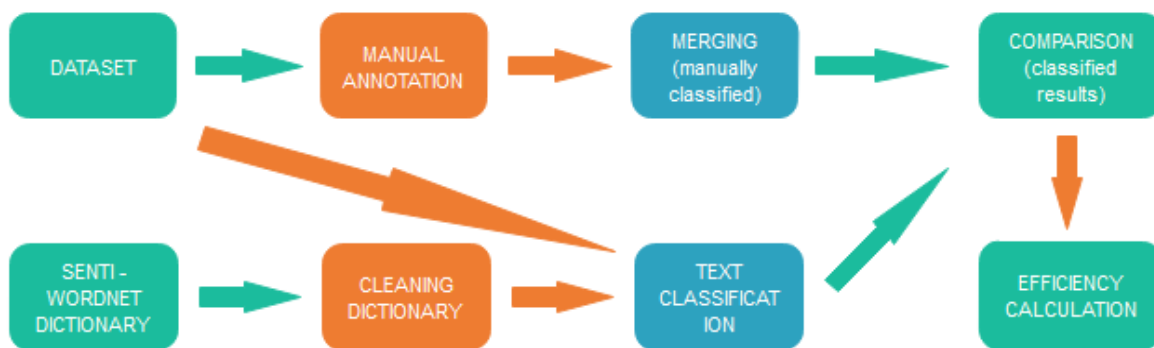
Hybrid-based approaches are roughly organized in three main components: normalization, lexicon-based classifier, and machine learning classifier. These components are connected in a pipeline architecture that extracts the best characteristics from each component. In this pipeline architecture, each classifier, in a sequential order, evaluates the Textual message. In each step, the classifier determines the polarity class of the message if a certain degree of confidence is achieved. Then the classifier in the next step is called. The machine learning classifier is the last step in the process and the positive results of lexicon analysis act as the training set for Machine learning models. It is responsible to determine the polarity even if the previous classifiers failed to achieve the confidence level required to classification. The normalization component is responsible to correct and normalize the text before the classifiers use it. This architecture improves the classification process because it takes advantage of the multiple approaches. For example, the rule-based classifier is the most reliable classifier. It achieves good results when the text is matched by a high-confidence rule. However, due to the freedom of language, rules may not match 100% of the unseen examples, consequently it has a low recall rate. Lexicon-based classifiers, for example, are very confident in the process to determine if a text is polar or neutral. Using sentiment lexicons, we can determine that sentences containing sentiment words are polar and sentences that do not contain such words are neutral. Moreover, the presence of a high number of positive or negative words in the text may be a strong indicative of the polarity. Finally, machine learning is known to be highly domain adaptive and to be able to find deep correlations. The last classifier might provide the final decision even when the previous methods failed.

Thus, a hybrid system encompasses the best of both worlds i.e., lexicon based and machine learning based and being a more-than-single stage process, only the true positives move on to the next iterations thus ensuring a high success rate.

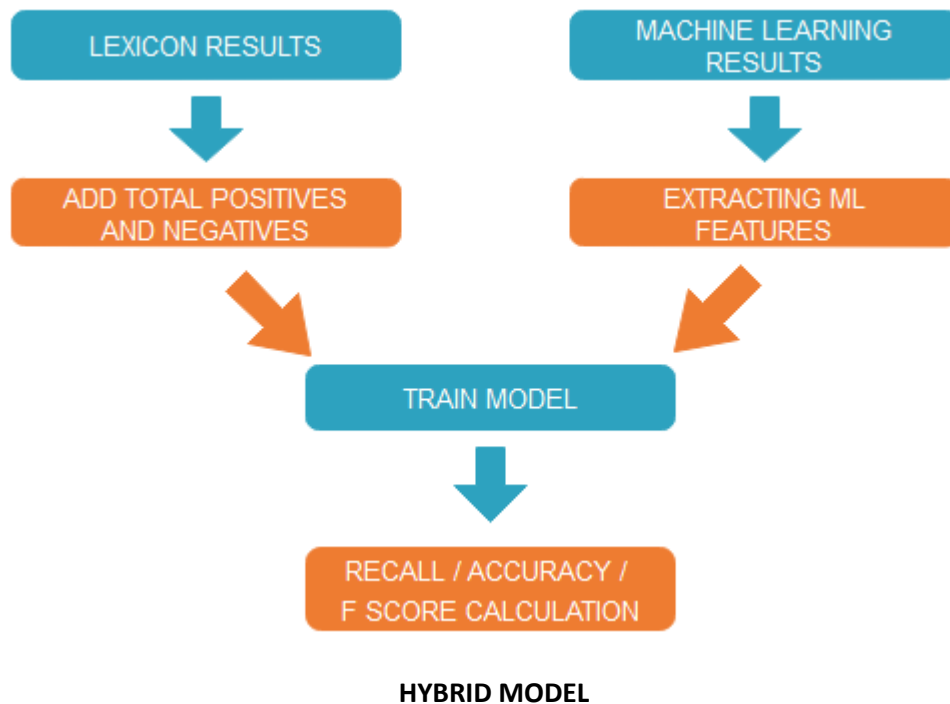
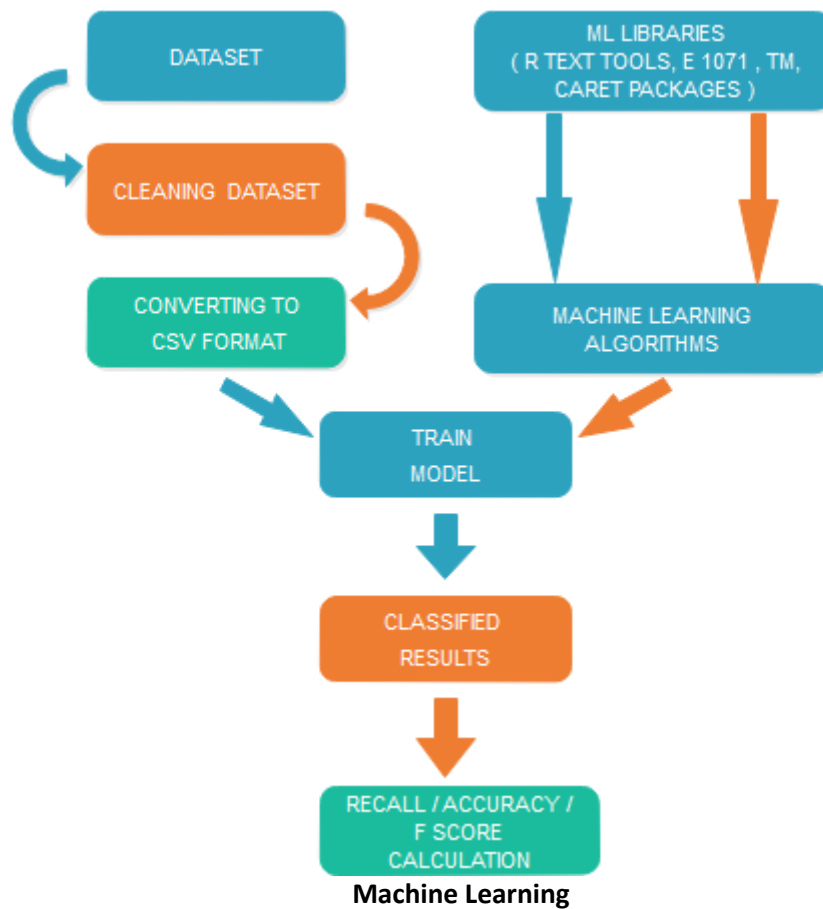
4. METHODOLOGY

4.1 Proposed System Architecture

The system performs Sentiment classification in three broad steps: (1) Manual annotation is applied on the Trip Advisor Dataset (3000+ hotel reviews) and the reviews are analyzed and classified by introducing a class tag under each review into five classes: Strongly Positive, Positive, Neutral, Negative and Strongly Negative. (2) A classifier is built with the vision of calculating efficiency using TRIE Data Structure(it is an information retrieval data structure) in Java that took the raw reviews as the inputs, used SentiWordNet (cleaned according to relevant requirements and suitable conditions) as the lexicon and classified the hotel reviews on basis of dictionary and scores in it. This step also works on determining a rating for the food and price aspect of the hotel on the basis of the reviews being analyzed. (3) The results obtained in first and second step are analyzed and compared and efficiency is calculated. There are also various sub steps included in each step.



LEXICAL ANALYSIS



This section includes the information about the Dataset that we used in our project work on sentiment analysis. We also propose the architecture of our system that we worked upon in the project. Each step of the methodology used/followed is explained.

4.2 The Dataset

We used the TripAdvisor Data Set for our research work in sentiment analysis. Parsed reviews used in our work are crawled from <http://www.tripadvisor.com>.

The Meta Data includes: Author, Content, Date, Number of Reader, Number of Helpful Judgment and Overall Rating. It is freely available at the web source: <http://times.cs.uiuc.edu/~wang296/Data/>.

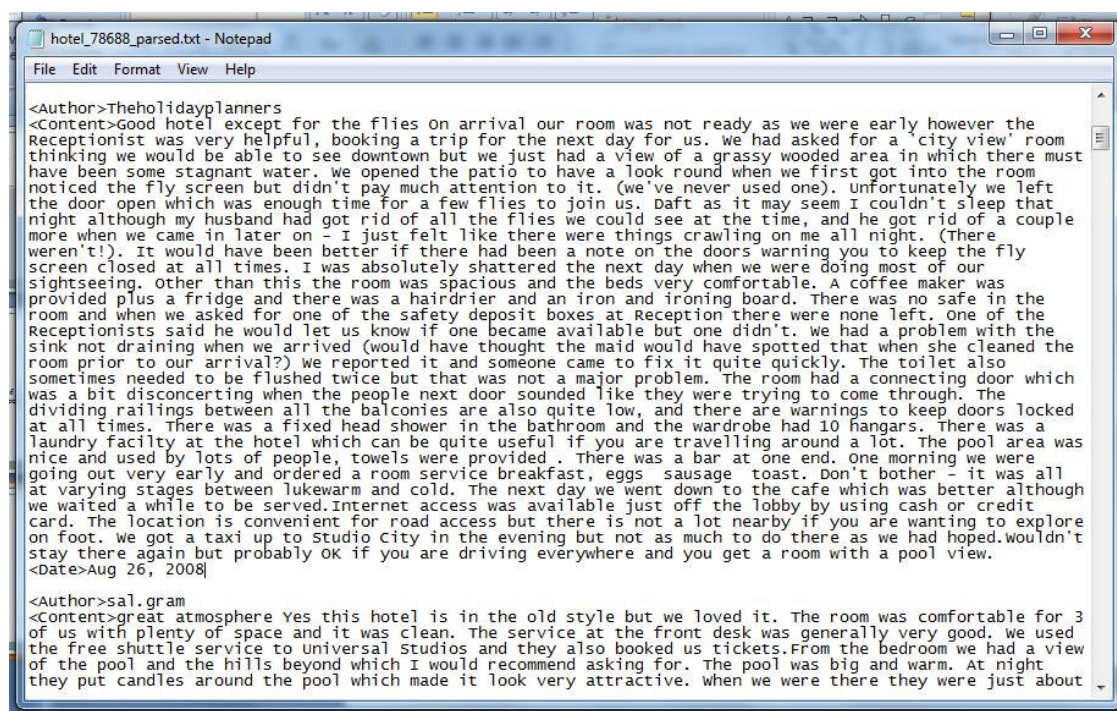


Fig. 4.1 One of the files of the dataset used

4.3 Manually Annotated Dataset

We introduced these 5 class tags in the dataset. After each review a tag according to the review was added for manual annotation. Tags used were:

<Class>Positive

<Class>StronglyPositive

<Class>Negative

<Class>StronglyNegative

<Class>Neutral

Figure 4.2 shows a section of one of the manually annotated review files.

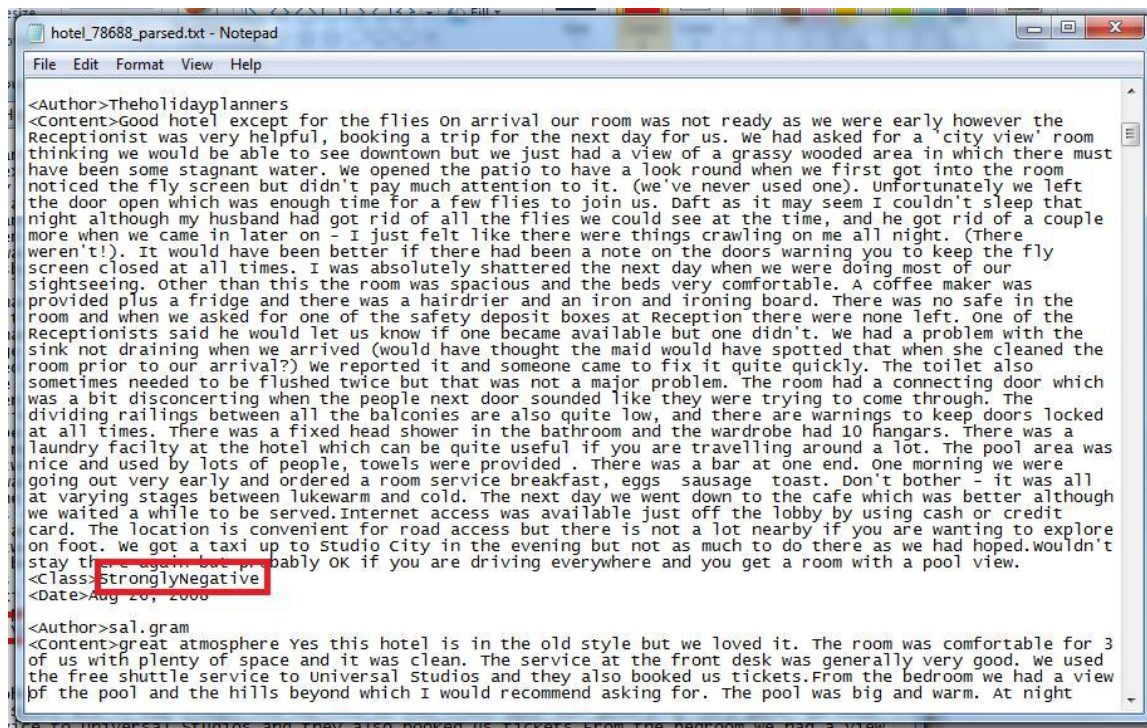


Fig. 4.2 One of the review files after manual annotation

We annotated 2500+ reviews manually.

❑ WE INTRODUCED THESE 4 TAGS IN THE DATSET . AFTER EACH REVIEW A TAG ACCORDING TO THE REVIEW WOULD BE ADDED FOR MANUAL CLSSIFICATION.

❑ The hotel will be judged on locality , food , price , service after reading the reviews on the rating of 1 to 5 , 1 being the worst and 5 the best.

- <Locality>
- <Food>
- <Price>
- <Service>

```
<Author>nzfligher
<Content>Hostel type accomodation My parents and I stayed in the family room for 4 nights in June 07. Upon arrival we were asked to
deposit $50 for 10 days which we weren't aware of (we booked throu Expedia.com.au and no such info was available). The room was small
which we didnt mind. The floor boards throughout the room made noise everytime you walked. The bed cover was faded ad had many stains
and a few holes. The beds are certainly not comfortable. One major issue we had was the Air Conditioning unit in the room was blowing
cold air in the middle of the night. Breakfast is quite simple and not worth $10. One thing that we were quite annoyed about was the
fact that the hotel is dirty by anyone's standard. The towels are well used and in need of replacement.In my opinion the hotel is not
a 3 star hotel. The hotel has vending machines for alcohol, bevereges,snacks, toothbrushes, etc... so you cannot really say that this
is a hotel, it had a hostel feel to it. I was aware that the hotel went through a soft refurbishing but I think soft refurbishing
meant that they changed the carpet only. Location is not very good, it takes over 10 mins to walk to central station or the Town Hall
station. The hotel is not too far to Darling Harbour which seems to be the only positive side of the hotel.Being on a main street, it
is very noisy at nights, there is a bar/pub downstairs which adds to the street noise! There are also restaurants on the other side o
the hotel and you can definitely hear the noise when you are n the hallways.I suggest to shop around for a hotel in sydney, this hote
is too expensive for $95-$115 a night for 2, mainly because it is a run down hotel, uncomfortable and lacks the wow factor. Hotels
around the Central Station are cheaper and in better condition, although it feels that they are not really in town but are only
seconds from the central station which means that for a day ticket you can go pretty much every where in town faster and with less
walking.
<Class>StronglyNegative
<Locality>2
<Food>2
<Price>1
<Service>1
<Date>Jun 11, 2007
<Rating>1      1      1      3      1      2      3      3
```

```
<Author>nzfligher
<Content>A Basic Old Hotel - Budget Hotel so don't expect anything! My parents and I stayed in the family room for 4 nights in June
07. Upon arrival we were asked to deposit $50 for 10 days which we weren't aware of (we booked throu Expedia.com.au and no such info
was available). The room was small which we didnt mind. The floor boards throughout the room made noise everytime you walked. The bed
cover was faded ad had many stains and a few holes. The beds are certainly not comfortable. One major issue we had was the Air
Conditioning unit in the room was blowing cold air in the middle of the night. Breakfast is quite simple and not worth $10. One thing
that we were quite annoyed about was the fact that the hotel is dirty by anyone's standard. The towels are well used and in need of
replacement.In my opinion the hotel is not a 3 star hotel. The hotel has vending machines for alcohol, bevereges,snacks, toothbrushes
etc... so you cannot really say that this is a hotel, it had a hostel feel to it. I was aware that the hotel went through a soft
refurbishing but I think soft refurbishing meant that they changed the carpet only. Location is not very good, it takes over 10 mins
to walk to central station or the Town Hall station. The hotel is not too far to Darling Harbour which seems to be the only positive
side of the hotel.Being on a main street, it is very noisy at nights, there is a bar/pub downstairs which adds to the street noise!
There are also restaurants on the other side of the hotel and you can definitely hear the noise when you are n the hallways.I suggest
to shop around for a hotel in sydney, this hotel is too expensive for $95-$115 a night for 2, mainly because it is a run down hotel,
uncomfortable and lacks the wow factor. Hotels around the Central Station are cheaper and in better condition, although it feels that
they are not really in town but are only seconds from the central station which means that for a day ticket you can go pretty much
```

Fig. 4.3 Review with tags

4.4 Developing the Classifier

Steps involved in developing the classifier are:

1. Cleaning the SentiWordNet dictionary.
2. Using Trie data structure to develop classifier and classifying the dataset using the classifier
3. Comparing the manual annotated results with the classifier results and calculating efficiency.

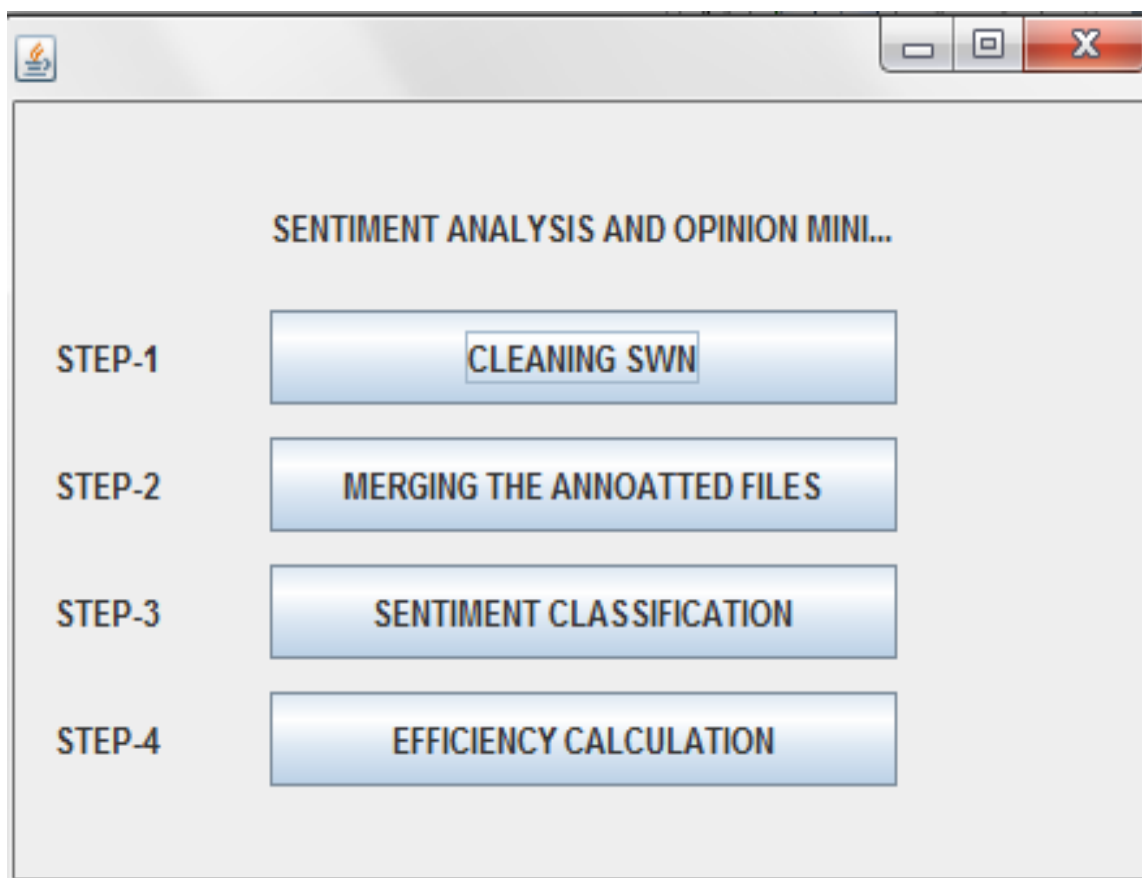
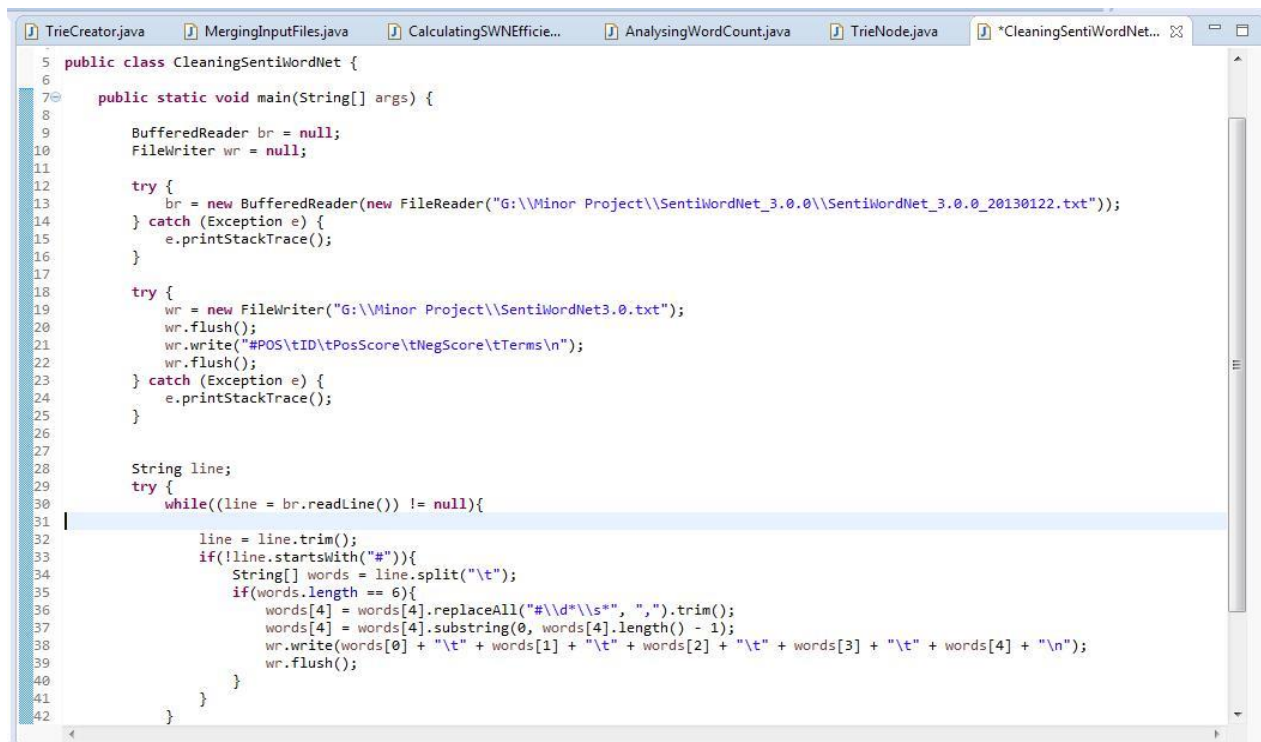


Fig. 4.4 Working of the classifier

Cleaning SentiWordNet

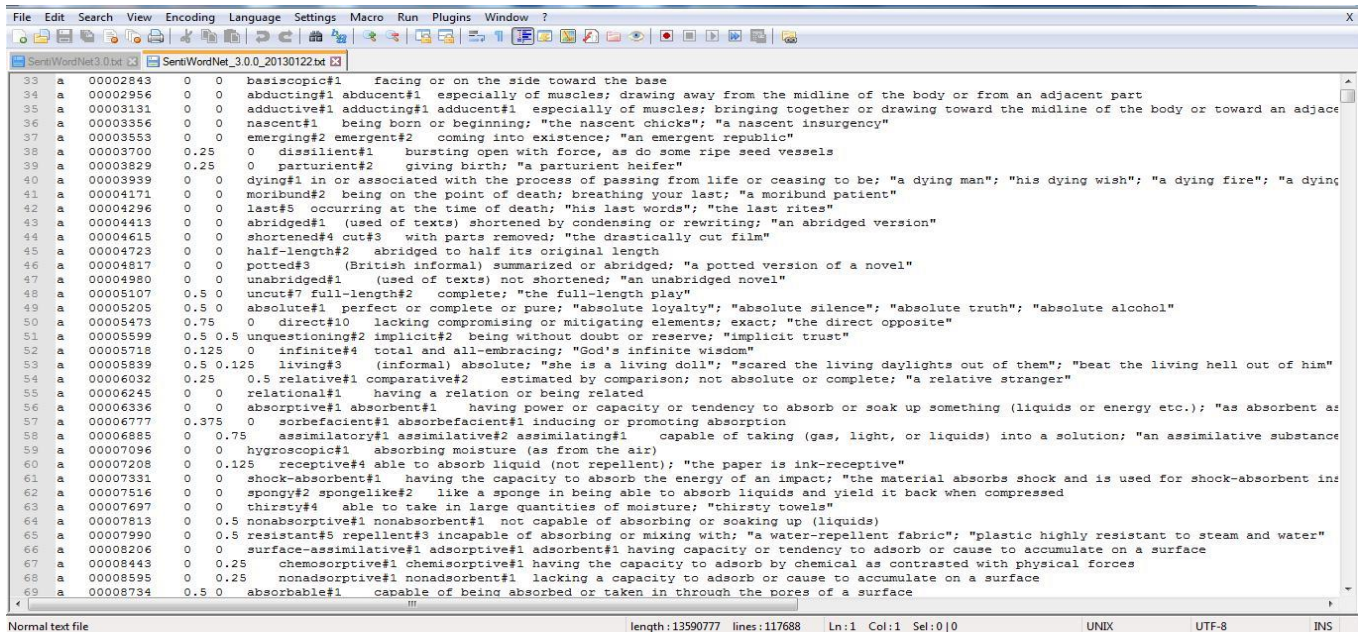
The lexicon resource being used, namely SentiWordNet, has a lot of information which was of very less or no use to the classifier which was built. So, before developing the classifier, the resource was cleaned in order to remove the useless part of the information.



```
5 public class CleaningSentiWordNet {
6
7     public static void main(String[] args) {
8
9         BufferedReader br = null;
10        FileWriter wr = null;
11
12        try {
13            br = new BufferedReader(new FileReader("G:\\Minor Project\\SentiWordNet_3.0.0\\SentiWordNet_3.0.0_20130122.txt"));
14        } catch (Exception e) {
15            e.printStackTrace();
16        }
17
18        try {
19            wr = new FileWriter("G:\\Minor Project\\SentiWordNet3.0.txt");
20            wr.flush();
21            wr.write("#POS\tID\tPosScore\tNegScore\tTerms\n");
22            wr.flush();
23        } catch (Exception e) {
24            e.printStackTrace();
25        }
26
27        String line;
28        try {
29            while((line = br.readLine()) != null){
30
31                line = line.trim();
32                if(!line.startsWith("#")){
33                    String[] words = line.split("\t");
34                    if(words.length == 6){
35                        words[4] = words[4].replaceAll("#\\d*\\s*", "").trim();
36                        words[4] = words[4].substring(0, words[4].length() - 1);
37                        wr.write(words[0] + "\t" + words[1] + "\t" + words[2] + "\t" + words[3] + "\t" + words[4] + "\n");
38                        wr.flush();
39                    }
40                }
41            }
42        }
```

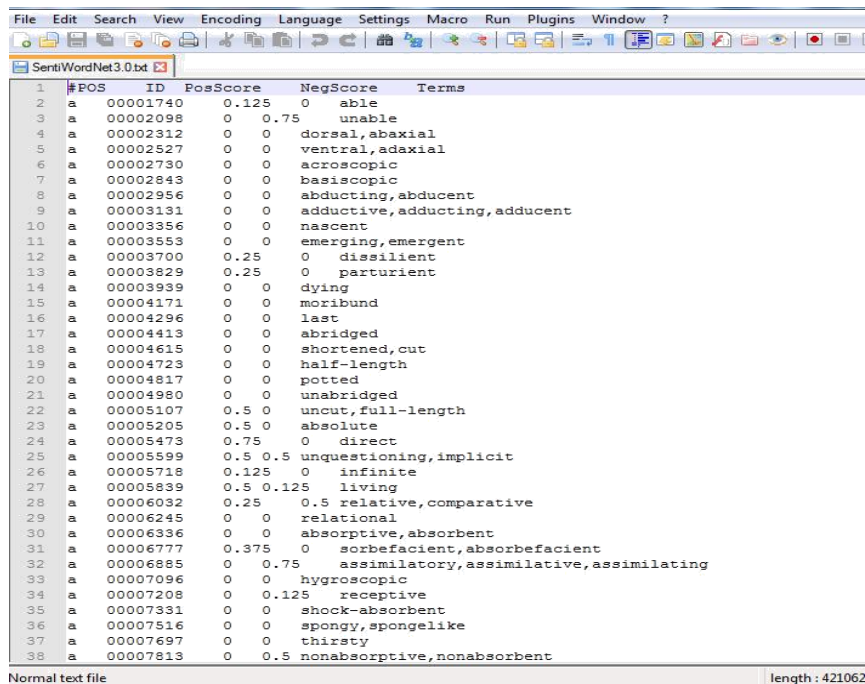
Fig. 4.5 A section of the code that was used to clean SentiWordNet

Figure 4.6 shows a section of the original SentiWordNet and Figure 4.7 shows a section of the cleaned SentiWordNet.



#	POS	ID	PosScore	NegScore	Terms
33	a	00002843	0	0	basiscopic#1 facing or on the side toward the base
34	a	00002956	0	0	abducting#1 abducent#1 especially of muscles; drawing away from the midline of the body or from an adjacent part
35	a	00003131	0	0	adductive#1 adducting#1 adducent#1 especially of muscles; bringing together or drawing toward the midline of the body or toward an adjacent part
36	a	00003356	0	0	nascent#1 being born or beginning; "the nascent chicks"; "a nascent insurgency"
37	a	00003553	0	0	emerging#2 emergent#2 coming into existence; "an emergent republic"
38	a	00003700	0.25	0	dissilient#1 bursting open with force, as do some ripe seed vessels
39	a	00003829	0.25	0	parturient#2 giving birth; "a parturient heifer"
40	a	00003939	0	0	dying#1 in or associated with the process of passing from life or ceasing to be; "a dying man"; "his dying wish"; "a dying fire"; "a dying
41	a	00004171	0	0	morbund#2 being on the point of death; breathing your last; "a moribund patient"
42	a	00004296	0	0	last#5 occurring at the time of death; "his last words"; "the last rites"
43	a	00004413	0	0	abridged#1 (used of texts) shortened by condensing or rewriting; "an abridged version"
44	a	00004615	0	0	shortened#4 cut#3 with parts removed; "the drastically cut film"
45	a	00004723	0	0	half-length#2 abridged to half its original length
46	a	00004817	0	0	potted#3 (British informal) summarized or abridged; "a potted version of a novel"
47	a	00004980	0	0	unabridged#1 (used of texts) not shortened; "an unabridged novel"
48	a	00005107	0.5	0	uncut#7 full-length#2 complete; "the full-length play"
49	a	00005205	0.5	0	absolute#1 perfect or complete or pure; "absolute loyalty"; "absolute silence"; "absolute truth"; "absolute alcohol"
50	a	00005473	0.75	0	direct#10 lacking compromising or mitigating elements; exact; "the direct opposite"
51	a	00005599	0.5	0.5	unquestioning#2 implicit#2 being without doubt or reserve; "implicit trust"
52	a	00005718	0.125	0	infinite#1 total and all-embracing; "God's infinite wisdom"
53	a	00005839	0.5	0.125	living#3 (informal) absolute; "she is a living doll"; "scared the living daylights out of them"; "beat the living hell out of him"
54	a	00006032	0.25	0.5	relative#1 comparative#2 estimated by comparison; not absolute or complete; "a relative stranger"
55	a	00006245	0	0	relational#1 having a relation or being related
56	a	00006336	0	0	absorptive#1 absorbent#1 having power or capacity or tendency to absorb or soak up something (liquids or energy etc.); "as absorbent as
57	a	00006777	0.375	0	sorbefacient#1 sorbefacient#2 inducing or promoting absorption
58	a	00006885	0	0.75	assimilatory#1 assimilative#2 assimilating#1 capable of taking (gas, light, or liquids) into a solution; "an assimilative substance
59	a	00007096	0	0	hygroscopic#1 absorbing moisture (as from the air)
60	a	00007208	0	0.125	receptive#4 able to absorb liquid (not repellent); "the paper is ink-receptive"
61	a	00007331	0	0	shock-absorbent#1 having the capacity to absorb the energy of an impact; "the material absorbs shock and is used for shock-absorbent in
62	a	00007516	0	0	spongy#2 spongelike#2 like a sponge in being able to absorb liquids and yield it back when compressed
63	a	00007697	0	0	thirsty#4 able to take in large quantities of moisture; "thirsty towels"
64	a	00007813	0	0.5	nonabsorptive#1 nonabsorbent#1 not capable of absorbing or soaking up (liquids)
65	a	00007990	0	0.5	resistant#3 repellent#3 incapable of absorbing or mixing with; "a water-repellent fabric"; "plastic highly resistant to steam and water"
66	a	00008206	0	0	surface-assimilative#1 adsorptive#1 adsorbent#1 having capacity or tendency to adsorb or cause to accumulate on a surface
67	a	00008443	0	0.25	chemisorptive#1 chemisorptive#1 having the capacity to adsorb by chemical as contrasted with physical forces
68	a	00008595	0	0.25	nonadsorptive#1 nonadsorbent#1 lacking a capacity to adsorb or cause to accumulate on a surface
69	a	00008794	0.5	0	absorbable#1 capable of being absorbed or taken in through the pores of a surface

Fig. 4.6 SentiWordNet (as it looked before cleaning)



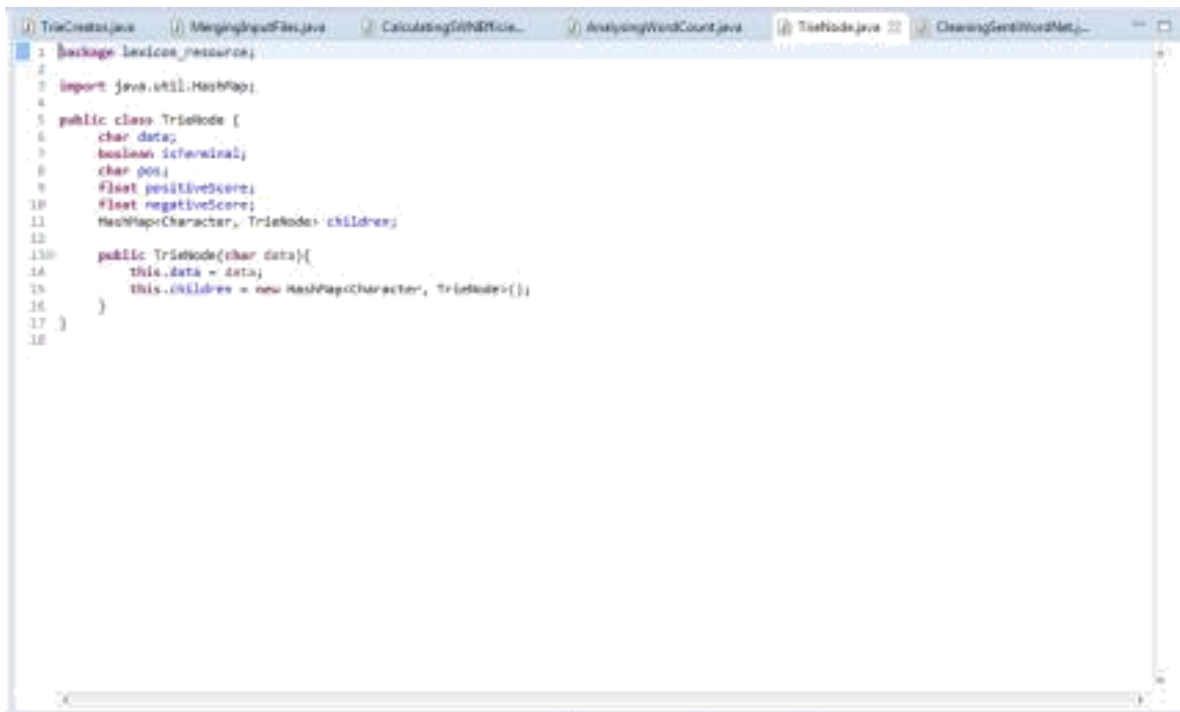
#	POS	ID	PosScore	NegScore	Terms
1	a	00001740	0.125	0	able
2	a	00002098	0	0.75	unable
3	a	00002312	0	0	dorsal, abaxial
4	a	00002527	0	0	ventral, adaxial
5	a	00002730	0	0	acroscopic
6	a	00002843	0	0	basiscopic
7	a	00002956	0	0	abducting, abducent
8	a	00003131	0	0	adductive, adducting, adducent
9	a	00003356	0	0	nascent
10	a	00003553	0	0	emerging, emergent
11	a	00003700	0.25	0	dissilient
12	a	00003829	0.25	0	parturient
13	a	00003939	0	0	dying
14	a	00004171	0	0	morbund
15	a	00004296	0	0	last
16	a	00004413	0	0	abridged
17	a	00004615	0	0	shortened, cut
18	a	00004723	0	0	half-length
19	a	00004817	0	0	potted
20	a	00004980	0	0	unabridged
21	a	00005107	0.5	0	uncut, full-length
22	a	00005205	0.5	0	absolute
23	a	00005473	0.75	0	direct
24	a	00005599	0.5	0.5	unquestioning, implicit
25	a	00005718	0.125	0	infinite
26	a	00005839	0.5	0.125	living
27	a	00006032	0.25	0.5	relative, comparative
28	a	00006245	0	0	relational
29	a	00006336	0	0	absorptive, absorbent
30	a	00006777	0.375	0	sorbefacient, sorbefacient
31	a	00006885	0	0.75	assimilatory, assimilative, assimilating
32	a	00007096	0	0	hygroscopic
33	a	00007208	0	0.125	receptive
34	a	00007331	0	0	shock-absorbent
35	a	00007516	0	0	spongy, spongelike
36	a	00007697	0	0	thirsty
37	a	00007813	0	0.5	nonabsorptive, nonabsorbent

Fig. 4.7 Cleaned SentiWordNet

Creating the Classifier and the Classification process

The Trie data structure (a data structure used for efficient information retrieval), was used for storing the lexicon resource for lookup during the classification process.

Figure 4.8 shows the Trie Node class and Figure 4.9 shows the code that creates the Trie to store the SentiWordNet resource.

A screenshot of an IDE window titled 'TrieNode.java 22'. The code is in Java and defines a 'TrieNode' class. The package is 'lexicon_resource'. It imports 'java.util.HashMap'. The class has attributes: 'char data', 'boolean isTerminal', 'char pos', 'float positiveScore', 'float negativeScore', and 'HashMap<Character, TrieNode> children'. The constructor 'TrieNode(char data)' initializes 'this.data = data' and 'this.children = new HashMap<Character, TrieNode>()'.

```
1 package lexicon_resource;
2
3 import java.util.HashMap;
4
5 public class TrieNode {
6     char data;
7     boolean isTerminal;
8     char pos;
9     float positiveScore;
10    float negativeScore;
11    HashMap<Character, TrieNode> children;
12
13    public TrieNode(char data){
14        this.data = data;
15        this.children = new HashMap<Character, TrieNode>();
16    }
17 }
18
```

Fig.4.8 Trie Node Class


```

38
39 private static Trie createTrie(){
40     Trie t = new Trie();
41     BufferedReader br = null;
42
43     try {
44         br = new BufferedReader(new FileReader("G:\\Minor Project\\SentiWordNet3.0.txt"));
45         br.readLine();
46     } catch (Exception e) {
47         e.printStackTrace();
48     }
49
50     String line;
51     try {
52         while((line = br.readLine()) != null){
53             String[] values = line.split("\t");
54
55             char pos = values[0].charAt(0);
56             float posScore = Float.parseFloat(values[2]);
57             float negScore = Float.parseFloat(values[3]);
58
59             String[] words = values[4].split(",");
60             int numberOfWords = words.length;
61
62             for(int i = 0; i < numberOfWords; i++){
63                 t.insert(words[i], pos, posScore, negScore);
64             }
65         }
66     } catch (Exception e) {
67         e.printStackTrace();
68     }
69
70     return t;
71 }
72
73 private static String getOutputFileName(String inputFileName){
74     StringBuffer sbuf = new StringBuffer();
75

```

Fig.4.9 Create Trie method of the Classifier

The classification process works by calculating and then comparing the total positive score of a review against its total negative score. Total Scores are calculated by adding the scores for each individual word.

Figure 4.10 shows the code that performs the classification process, and Figure 4.11 shows the file containing the classifications.

```

86 private static void classify(Trie t, File file){
87     if(file == null)
88         System.exit(0);
89
90     BufferedReader br = null;
91     FileWriter wr = null;
92     String line;
93     try {
94         br = new BufferedReader(new FileReader(file));
95         wr = new FileWriter(getOutputFileName(file.getAbsolutePath()));
96         wr.flush();
97         wr.write("#AnnotatedClass\tClassification\n");
98         wr.flush();
99     } catch (Exception e) {
100         e.printStackTrace();
101     }
102
103     int totalFoodWords = 0;
104     int totalPriceWords = 0;
105     float positiveFoodScore = 0.0f;
106     float negativeFoodScore = 0.0f;
107     float positivePriceScore = 0.0f;
108     float negativePriceScore = 0.0f;
109
110     try {
111         while((line = br.readLine()) != null){
112             if(line.startsWith("<Content>")){
113                 totalReviews++;
114
115                 line = line.substring(9).trim();
116                 line = line.replaceAll("\\p{Punct}", "");
117                 String[] words = line.split(" ");
118                 int numberOfWords = words.length;
119
120                 float positiveValue = 0.0f;
121                 float negativeValue = 0.0f;
122
123                 for(int i = 0; i < numberOfWords; i++){

```

Fig.4.10 Classify method of the Classifier

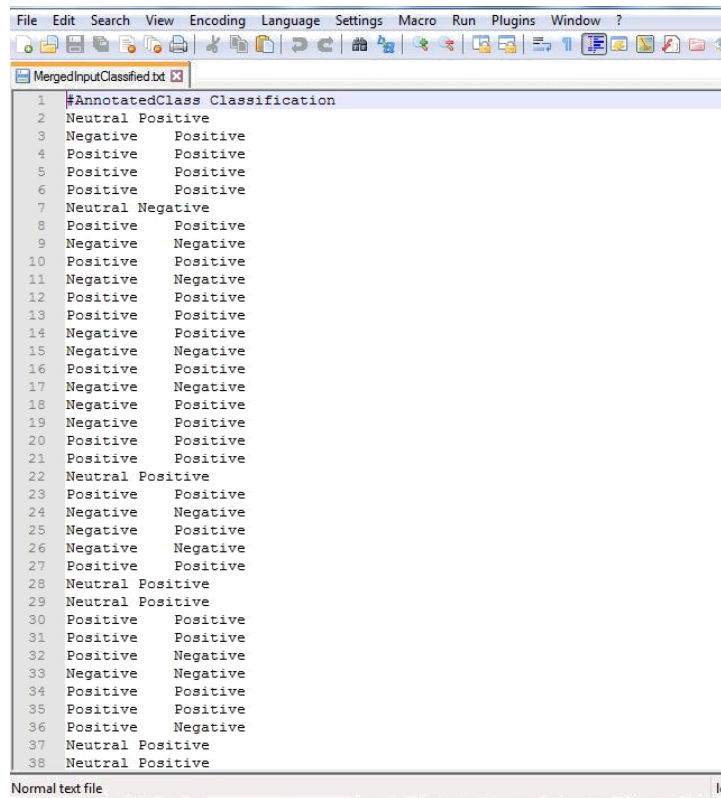


Fig.4.11 Classified File (the file containing classifications)

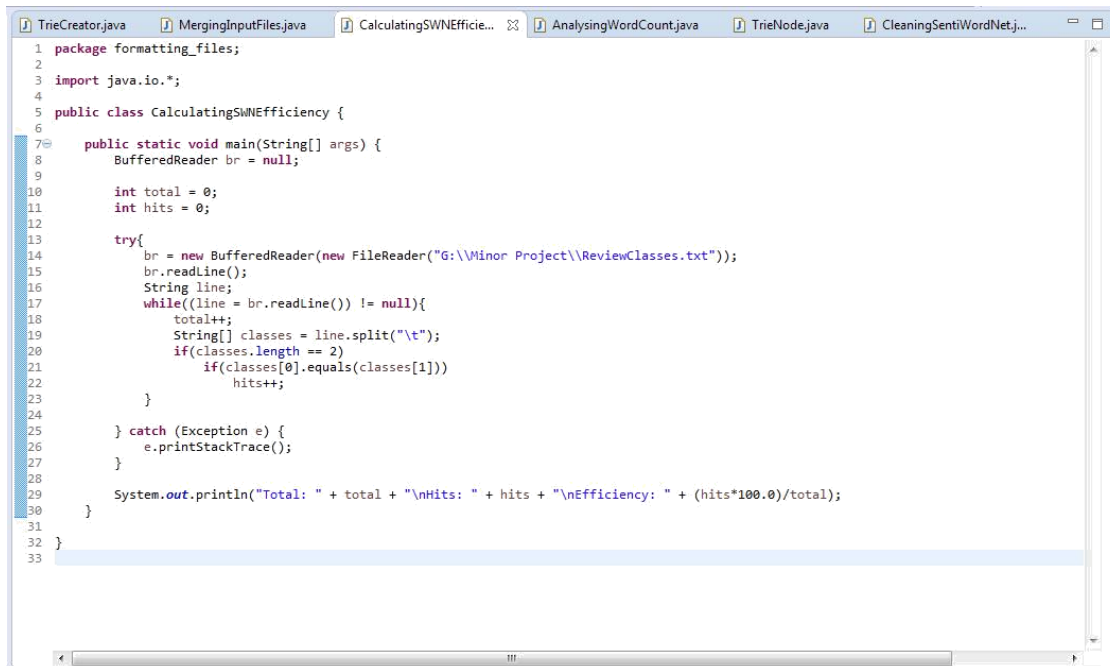
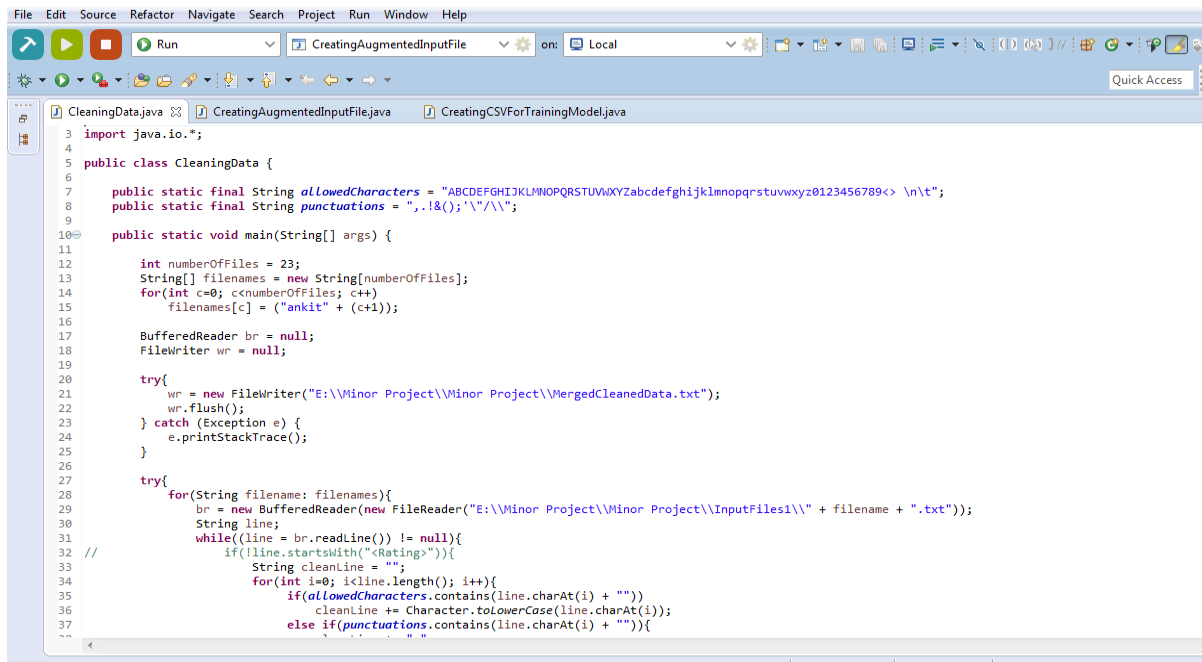


Fig.4.12 Calculating SWN efficiency

MACHINE LEARNING OF HOTEL REVIEWS

```
1 <Author>Durhamguy
2 <Content>Not the best and not a boutique hotel Stayed at the Capitol Hotel as my last stop in Australia, loved Sydney but not
this hotel. It is a basic three star and the fact that it is owned by Rydges is quite surprising. Definety not a boutique
hotel unless this is the ambition. Certainly the work and transformation had not started during my stay in July.First
impression was not good the reception area has a sterile youth hostel feel about it. The girl on reception was okay but no
Aussie welcome. During my stay I would say some staff here where polite and friendly while others had little motivation in
their job. Stayed in room 514, which was compact and comfortbale but nothing special. Bathroom very small and quite
tacky.Location wise no complaints, caught the train from the airport to Central Station which is about a 10 - 15 minute
journey then its just a short walk to the hotel. Capitol Square is right next to the theatre of the same name with a small
shopping centre, Starbucks and Irish pub in the same block. Directly next to China Town some of which isn't that picturesque,
but Darling Harbour which is brilliant is only a 15 minute walk away.Circular Quay is easy to reach either by taking the
train, or a ferry from Darling Harbour which is certainly the most fun way or a good walk down George Street.Never had
breakfast in the hotel instead whent along to Darling Harbour and had great meals sitting in the sunshine in one of the
numerous outdoor cafes.
3 <Class>Positive
4 <Locality>4
5 <Food>0
6 <Price>3
7 <Service>4
8 <Date>Nov 22, 2008
9 <Rating>2 2 3 4 3 2 2 3
10
11 <Author>tripp0ne
12 <Content>Noisy. I requested a quiet room when booking. I was given a room where you could distinctly hear people in the
adjacent rooms coughing or talking, which make me wake up at nighttime. The room was tiny, but this would have been fine for
the price I paid, if a better sound insulation had been provided. This is not what you expect from a hotel of this category.
No information at all was provided about telephone and internet rates in the room. I came back one night and found the
entrance shut with a sign saying reception is temporarily closed, and I had to wait outside for a while before being allowed
to get into my room.
13 <Class>StronglyNegative
14 <Locality>0
15 <Food>0
```

Fig.4.13 MANUALLY ANNOTATED DATASE



```
File Edit Source Refactor Navigate Search Project Run Window Help
Run CreatingAugmentedInputFile on: Local
CleaningData.java CreatingAugmentedInputFile.java CreatingCSVForTrainingModel.java
3 import java.io.*;
4
5 public class CleaningData {
6
7     public static final String allowedCharacters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789<> \\n\\t";
8     public static final String punctuations = ",.;&();'\"";
9
10    public static void main(String[] args) {
11
12        int numberOfFiles = 23;
13        String[] filenames = new String[numberOfFiles];
14        for(int c=0; c<numberOfFiles; c++){
15            filenames[c] = ("ankit" + (c+1));
16
17            BufferedReader br = null;
18            FileWriter wr = null;
19
20            try{
21                wr = new FileWriter("E:\\Minor Project\\Minor Project\\MergedCleanedData.txt");
22                wr.flush();
23            } catch (Exception e) {
24                e.printStackTrace();
25            }
26
27            try{
28                for(String filename: filenames){
29                    br = new BufferedReader(new FileReader("E:\\Minor Project\\Minor Project\\InputFiles\\\" + filename + ".txt"));
30                    String line;
31                    while((line = br.readLine()) != null){
32                        if(!line.startsWith("<Rating:>")){
33                            String cleanLine = "";
34                            for(int i=0; i<line.length(); i++){
35                                if(allowedCharacters.contains(line.charAt(i) + ""))
36                                    cleanLine += Character.toLowerCase(line.charAt(i));
37                                else if(punctuations.contains(line.charAt(i) + "")){
38                                    //
39                                }
40                            }
41                        }
42                    }
43                }
44            } catch (Exception e) {
45                e.printStackTrace();
46            }
47        }
48    }
49 }
```

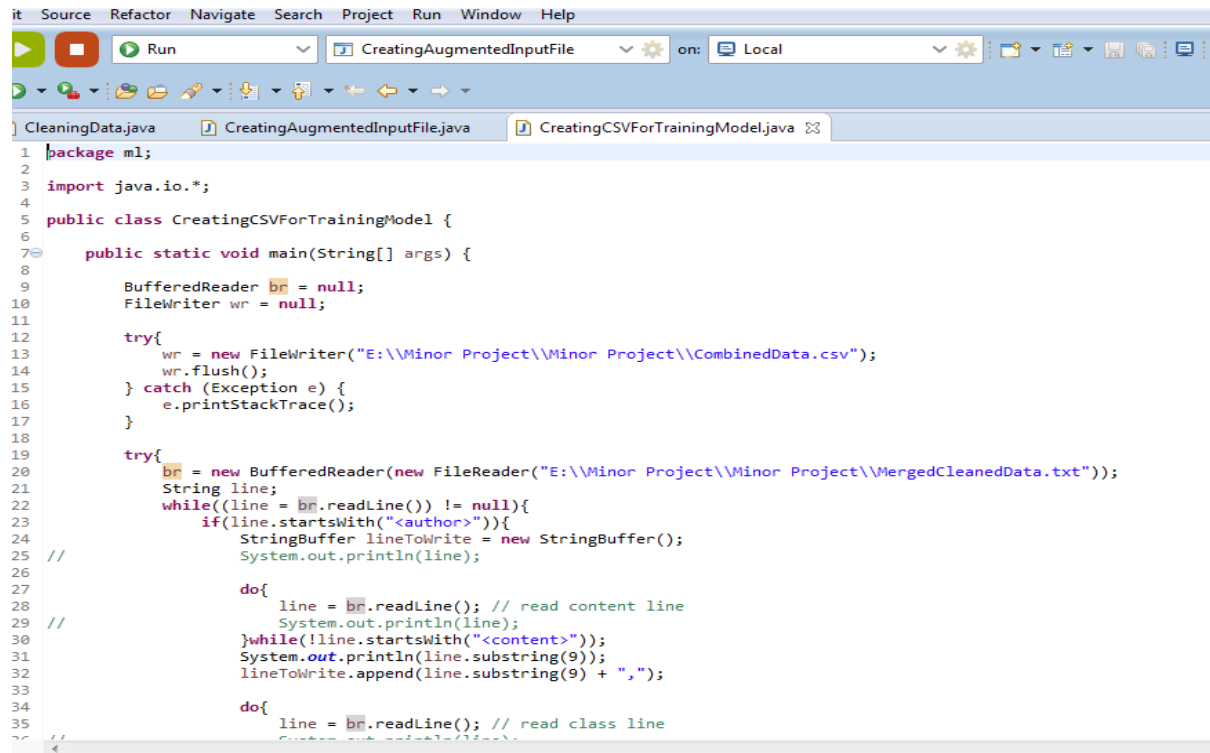
Fig.4.14 CLEANING DATASET

```

1 <author>fantakil
2 <content>good location affordable price one bad surprise we were looking for an affordable place to stay in boston and we got ar
3 <class>neutral
4 <locality>4
5 <food>3
6 <price>3
7 <service>4
8 <date>nov 27 2008
9 <rating>3 4 4 5 1 4 1 3
10
11 <author>bittern
12 <content>way out in boston best western terrace inn 1650 commonwealth avenue boston ma usastayed sep 2008 one night double room
13 <class>stronglynegative
14 <locality>2
15 <food>2
16 <price>2
17 <service>4
18 <date>sep 18 2008
19 <rating>3 3 4 2 4 3 1 1
20
21 <author>lovelybike
22 <content>some very good points first of all it was well situated for us we took the tram subway to visit downtown we felt quite s
23 <class>positive
24 <locality>4
25 <food>4
26 <price>3
27 <service>4
28 <date>aug 26 2008
29 <rating>4 4 3 4 4 5 4 4
30

```

Fig.4.15 CLEANED DATASET



```

it Source Refactor Navigate Search Project Run Window Help
Run CreatingAugmentedInputFile on: Local
CleaningData.java CreatingAugmentedInputFile.java CreatingCSVForTrainingModel.java
1 package ml;
2
3 import java.io.*;
4
5 public class CreatingCSVForTrainingModel {
6
7     public static void main(String[] args) {
8
9         BufferedReader br = null;
10        FileWriter wr = null;
11
12        try{
13            wr = new FileWriter("E:\\Minor Project\\Minor Project\\CombinedData.csv");
14            wr.flush();
15        } catch (Exception e) {
16            e.printStackTrace();
17        }
18
19        try{
20            br = new BufferedReader(new FileReader("E:\\Minor Project\\Minor Project\\MergedCleanedData.txt"));
21            String line;
22            while((line = br.readLine()) != null){
23                if(line.startsWith("<author>")){
24                    StringBuffer lineToWrite = new StringBuffer();
25                    // System.out.println(line);
26
27                    do{
28                        line = br.readLine(); // read content line
29                        System.out.println(line);
30                    }while(!line.startsWith("<content>"));
31                    System.out.println(line.substring(9));
32                    lineToWrite.append(line.substring(9) + "," );
33
34                    do{
35                        line = br.readLine(); // read class line
36                        System.out.println(line);

```

Fig.4.16 CREATING CSV FOR TRAINING MODEL

	A	B	C	D	E	F	G
1	good location affordable price one bad surprise we were looking for an affordable place to stay in bost	neutral	4	3	3	4	
2	way out in boston best western terrace inn 1650 commonwealth avenue boston ma usastayed sep 2008	negative	2	2	2	4	
3	pleasantly surprised i ve been by this hotel many times since i live in the area i always thought it look	neutral	3	3	3	3	
4	awful we found this hotel through a college website it was listed as a hotel in the vicinity of boston coll	negative	1	0	1	1	
5	careful its cheap but shabby nice staff but the place is well past its best when we checked in our initial	negative	2	0	2	3	
6	awful we booked this hotel because the boston college website listed it as a local place to stay we plan	negative	1	0	2	1	
7	ok but some cons i stayed at this hotel for 2 nights in october the hotel was very clean seemed to be in	negative	3	0	2	1	
8	great area not great hotel i ended up getting this hotel through priceline there was a conference in tow	negative	4	1	1	3	
9	not what we expected we booked this hotel because it was very close to boston university where we w	negative	4	2	1	1	
10	not the best best western we stayed here with my 14 month old daughter for a week in june of 2006 th	negative	2	0	2	2	
11	good location with free parking but not a favorite we spent 2 nights at the terrace inn it is located in a v	neutral	4	0	0	2	
12	pleasantly surprised i was pleasantly surprised at the great service location and price of this hotel with	positive	4	4	4	4	
13	never again don t even think about it don t go there don t stay there avoid it like the plaguebest wester	negative	0	0	0	0	
14	2 rooms out of 3 not bad the terrace inn is very convenient is well maintained has a reasonable breakf	negative	3	3	3	2	
15	totally gross and cheesy let me first admit that i m giving this an ok rating due to the fact that my wifi w	negative	0	0	2	1	
16	eh stayed here overnight 3 113 12 first the positives close to a t stop about a block awaybrookline area i	neutral	4	2	4	2	
17	best feature of this hotel is their people we found this hotel to be very convenient and clean as well as	neutral	4	0	0	5	
18	good hotel for your i stayed here for 7 days and was pleasantly surprised especially after reading some	positive	5	4	4	4	
19	roomy suite could be cleaner the suites at the best western terrace inn are a decent size with plenty of	positive	4	3	0	2	
20	good experience with the extended stay package i stayed here in september of 2004 for 1 month their	positive	3	3	4	4	
21	notsogreat stay we stayed at the terrace inn in the summer of 04 the hotel is in an awkward location an	negative	2	1	0	1	
22	conveniently situated but expensive not having been to boston before i chose this hotel because it wa	neutral	4	3	2	4	
23	good location to local hospitals we stayed one night at the inn at longwood due to its close proximately	neutral	4	3	3	4	
24	decent place would recommend this best western is nicer and cleaner than most best westerns initiall	negative	3	4	3	3	
25	gives new meaning to the word filthy my husband and i stayed at the inn at longwood medical becaus	negative	4	3	2	1	

Fig.4.17 CSV FILE OF DATASET

```

1 public class CreatingAugmentedInputFile {
2
3     public static void main(String[] args) {
4         Trie t = Classifier.createTrie();
5         augmentScoresToReviewsTrainVersion(t);
6     }
7
8     public static void augmentScoresToReviewsTrainVersion(Trie t){
9         BufferedReader br = null;
10        FileWriter wr = null;
11        String line;
12        try {
13            br = new BufferedReader(new FileReader("E:\\Minor Project\\Minor Project\\CombinedData.csv"));
14            wr = new FileWriter("E:\\Minor Project\\Minor Project\\AugmentedTrainingDataTemp.csv");
15            wr.flush();
16        } catch (Exception e) {
17            e.printStackTrace();
18        }
19
20        try {
21            while((line = br.readLine()) != null){
22                String[] values = line.split(",");
23                if(values.length >= 6){
24                    String[] words = values[0].split(" ");
25                    int numberOfWords = words.length;
26
27                    float positiveScore = 0.0f;
28                    float negativeScore = 0.0f;
29
30                    for(int i = 0; i < numberOfWords; i++){
31                        BooleanTwoFloatsChar thisWord = t.contains(words[i].trim().toLowerCase());
32                        positiveScore += thisWord.x;
33                        negativeScore += thisWord.y;
34                    }
35                }
36            }
37        }
38    }
39 }

```

Fig. 4.18 ADDING AVERAGE POSITIVE & NEGATIVE SCORES IN DATASET

	A	B	C	D	E	F	G	H	I
1	good location affordable price one bad surprise we were looking for an affordable place t	0.06875	0.041875	neutral	4	3	3	4	
2	way out in boston best western terrace inn 1650 commonwealth avenue boston ma usast	0.037744	0.03125	negative	2	2	2	4	
3	pleasantly surprised i ve been by this hotel many times since i live in the area i always thc	0.040471	0.055328	neutral	3	3	3	3	
4	awful we found this hotel through a college website it was listed as a hotel in the vicinity	0.031145	0.0383	negative	1	0	1	1	
5	careful its cheap but shabby nice staff but the place is well past its best when we checked	0.074438	0.077247	negative	2	0	2	3	
6	awful we booked this hotel because the boston college website listed it as a local place tc	0.042051	0.035714	negative	1	0	2	1	
7	ok but some cons i stayed at this hotel for 2 nights in october the hotel was very clean see	0.045858	0.046598	negative	3	0	2	1	
8	great area not great hotel i ended up getting this hotel through priceline there was a confi	0.033221	0.0625	negative	4	1	1	3	
9	not what we expected we booked this hotel because it was very close to boston universit	0.044521	0.027397	negative	4	2	1	1	
10	not the best best western we stayed here with my 14 month old daughter for a week in ju	0.059211	0.051809	negative	2	0	2	2	
11	good location with free parking but not a favorite we spent 2 nights at the terrace inn it is	0.037946	0.03125	neutral	4	0	0	2	
12	pleasantly surprised i was pleasantly surprised at the great service location and price of th	0.071429	0.021825	positive	4	4	4	4	
13	never again don t even think about it don t go there don t stay there avoid it like the plagu	0.025	0.07	negative	0	0	0	0	
14	2 rooms out of 3 not bad the terrace inn is very convenient is well maintained has a reas	0.03149	0.030048	negative	3	3	3	2	
15	totally gross and cheesy let me first admit that i m giving this an ok rating due to the fact tl	0.059677	0.062903	negative	0	0	2	1	
16	eh stayed here overnight 3 113 12 first the positives close to a t stop about a block awaybr	0.047733	0.037825	neutral	4	2	4	2	
17	best feature of this hotel is their people we found this hotel to be very convenient and cl	0.060315	0.02972	neutral	4	0	0	5	
18	good hotel for your i stayed here for 7 days and was pleasantly surprised especially after i	0.055556	0.037247	positive	5	4	4	4	
19	roomy suite could be cleaner the suites at the best western terrace inn are a decent size v	0.045833	0.0125	positive	4	3	0	2	
20	good experience with the extended stay package i stayed here in september of 2004 for 1	0.033245	0.042553	positive	3	3	4	4	
21	notsogreat stay we stayed at the terrace inn in the summer of 04 the hotel is in an awkwar	0.027322	0.034153	negative	2	1	0	1	
22	conveniently situated but expensive not having been to boston before i chose this hotel k	0.081019	0.046296	neutral	4	3	2	4	
23	good location to local hospitals we stayed one night at the inn at longwood due to its closi	0.048119	0.03042	neutral	4	3	3	4	
24	decent place would recommend this best western is nicer and cleaner than most best we	0.044643	0.037338	negative	3	4	3	3	
25	glives new meaning to the word filthy my husband and i stayed at the inn at longwood m	0.055405	0.07027	negative	4	3	2	1	

Fig.4.19 HYBRID DATASET

APPLYING MACHINE LEARNING TECHNIQUE ON DATASET & HYBRID DATASET

The image displays three sequential screenshots of the RStudio environment, illustrating the application of different machine learning techniques to a dataset.

Top Screenshot: Naive Bayes Technique

```

1 # Include the required libraries
2 library(RTextTools)
3 library(e1071)
4 library(tm)
5 library(caret)
6
7
8 # reading data into data frames
9 reviews = read.csv("CleanedCombinedDataCompressed.csv", header = FALSE, fill = TRUE)
10 augmented.reviews = read.csv("AugmentedTrainingData.csv", header = FALSE, fill = TRUE)
11
12 # build dtm
13 mat= create_matrix(reviews[,1], language="english",
14                   removestopwords=FALSE, removeNumbers=TRUE,
15                   stemwords=TRUE)
16 sparse <- removeSparseTerms(mat, 0.7)
17 matrix = as.matrix(sparse)
18
19 aug.mat= create_matrix(augmented.reviews[,1], language="english",
20                      removestopwords=FALSE, removeNumbers=TRUE,
21                      stemwords=TRUE)
22 sparse <- removeSparseTerms(aug.mat, 0.81)
23 aug.matrix = as.matrix(sparse)
24 augmented.matrix = cbind(aug.matrix[,], augmented.reviews[,2], augmented.reviews[,3])
25
26
27 # Naive Bayes Technique
28
29 classifier = naiveBayes(matrix[1:800,], as.factor(reviews[1:800,2]), laplace = 1 )
30 predicted = predict(classifier, matrix[801:1192,])
31 table(reviews[801:1192, 2], predicted)
32 recall_accuracy(reviews[801:1192, 2], predicted)
33 conf.mat <- confusionMatrix(predicted, reviews[801:1192,2])
34 conf.mat$overall["Accuracy"]
35
36
37
38
39
40
41
42
43
44
45
46
47 # build container for training models and training models
48 container = create_container(matrix, as.numeric(as.factor(reviews[,2])),
49                             trainSize=1:800, testSize=801:1192, virgin=FALSE)
50 models = train_models(container, algorithms=c("SVM", "RF"))
51 results = classify_models(container, models)
52
53 # augmented.container = create_container(augmented.matrix, as.numeric(as.factor(augmented.reviews[,2])),
54                                       trainSize=1:800, testSize=801:1192, virgin=FALSE)
55 augmented.models = train_models(augmented.container, algorithms=c("SVM", "RF"))
56 augmented.results = classify_models(augmented.container, augmented.models)
57
58
59 # Support Vector Machine Technique
60 table(as.factor(reviews[801:1192, 2]), results[, "SVM_LABEL"])
61

```

Environment Panel:

Object	Class	Size
aug.matrix	Large matrix	(27685 elements, 748.2 kb)
augmented.matr...	Large matrix	(29267 elements, 760.4 kb)
augmented.resu...	392 obs. of 4 variables	
augmented.revi...	791 obs. of 8 variables	
mat	Large matrix	(16611 elements, 660.9 kb)
results	392 obs. of 4 variables	
reviews	791 obs. of 6 variables	

Bottom Screenshot: Random Forest Technique

```

67 set.seed(2014)
68 cross_validate(augmented.container, N, "SVM")
69
70 # Random Forest Technique
71 table(as.factor(reviews[801:1192, 2]), results[, "FORESTS_LABEL"])
72 #recall_accuracy(as.numeric(as.factor(reviews[801:1192, 4])), results[, "FORESTS_LABEL"])
73 N=4
74 set.seed(2014)
75 cross_validate(container, N, "RF")
76
77 table(as.factor(augmented.reviews[801:1192, 4]), augmented.results[, "FORESTS_LABEL"])
78 #recall_accuracy(as.numeric(as.factor(augmented.reviews[801:1192, 4])), augmented.results[, "FORESTS_LABEL"])
79 N=4
80 set.seed(2014)
81 cross_validate(augmented.container, N, "RF")
82
83
84 # Naive Bayes revisited without Neutral Class
85 reviews = read.csv("CleanedCombinedDataWithoutNeutralCompressed.csv", header = FALSE, fill = TRUE)
86 matrix= create_matrix(reviews[,1], language="english",
87                      removestopwords=FALSE, removeNumbers=TRUE,
88                      stemwords=TRUE)
89 sparse <- removeSparseTerms(matrix, 0.6)
90 mat = as.matrix(sparse)
91
92 classifier = naiveBayes(mat[1:600,], as.factor(reviews[1:600,2]), laplace = 1 )
93 predicted = predict(classifier, mat[601:791,])
94 table(reviews[601:791, 2], predicted)
95 recall_accuracy(reviews[601:791, 2], predicted)
96 conf.mat <- confusionMatrix(predicted, reviews[601:791,2])
97 conf.mat$overall["Accuracy"]
98
99 augmented.reviews = read.csv("AugmentedTrainingDataWithoutNeutral.csv", header = FALSE, fill = TRUE)
100 aug.mat= create_matrix(augmented.reviews[,1], language="english",
101                      removestopwords=FALSE, removeNumbers=TRUE,
102                      stemwords=TRUE)
103 sparse <- removeSparseTerms(aug.mat, 0.81)
104 aug.matrix = as.matrix(sparse)
105 augmented.matrix = cbind(aug.matrix[,], augmented.reviews[,2], augmented.reviews[,3])
106
107
108 # Naive Bayes Technique
109
110 classifier = naiveBayes(matrix[1:800,], as.factor(reviews[1:800,2]), laplace = 1 )
111 predicted = predict(classifier, matrix[801:1192,])
112 table(reviews[801:1192, 2], predicted)
113 recall_accuracy(reviews[801:1192, 2], predicted)
114 conf.mat <- confusionMatrix(predicted, reviews[801:1192,2])
115 conf.mat$overall["Accuracy"]
116
117
118 # build container for training models and training models
119 container = create_container(matrix, as.numeric(as.factor(reviews[,2])),
120                             trainSize=1:800, testSize=801:1192, virgin=FALSE)
121 models = train_models(container, algorithms=c("SVM", "RF"))
122 results = classify_models(container, models)
123
124 # augmented.container = create_container(augmented.matrix, as.numeric(as.factor(augmented.reviews[,2])),
125                                       trainSize=1:800, testSize=801:1192, virgin=FALSE)
126 augmented.models = train_models(augmented.container, algorithms=c("SVM", "RF"))
127 augmented.results = classify_models(augmented.container, augmented.models)
128
129 # Support Vector Machine Technique
130 table(as.factor(reviews[801:1192, 2]), results[, "SVM_LABEL"])
131

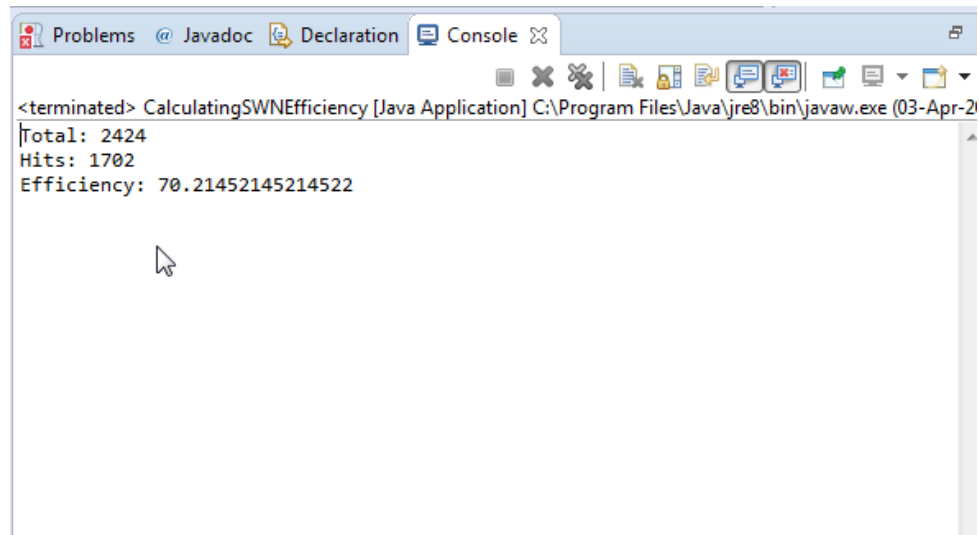
```

Environment Panel:

Object	Class	Size
aug.matrix	Large matrix	(27685 elements, 748.2 kb)
augmented.matr...	Large matrix	(29267 elements, 760.4 kb)
augmented.resu...	392 obs. of 4 variables	
augmented.revi...	791 obs. of 8 variables	
mat	Large matrix	(16611 elements, 660.9 kb)
results	392 obs. of 4 variables	
reviews	791 obs. of 6 variables	

Comparing Results and Calculating Efficiency

The reviews file was created such that it contains both the classifier's classification along with the manual annotation. The reviews file was used to calculate the number of hits for the classifier and its efficiency was calculated.



```
<terminated> CalculatingSWNEfficiency [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (03-Apr-2018)
Total: 2424
Hits: 1702
Efficiency: 70.21452145214522
```

Fig.4.20 EFFICIENCY OF LEXICON CLASSIFIER

RECALL ACCURACIES

	NAIVE BAYES	SUPPORT VECTOR MACHINE	RANDOM FOREST
MACHINE LEARNING	0.5790	0.6954	0.7279
HYBRID	0.6122	0.6974	0.7755

Fig.4.21 RECALL ACCURACY

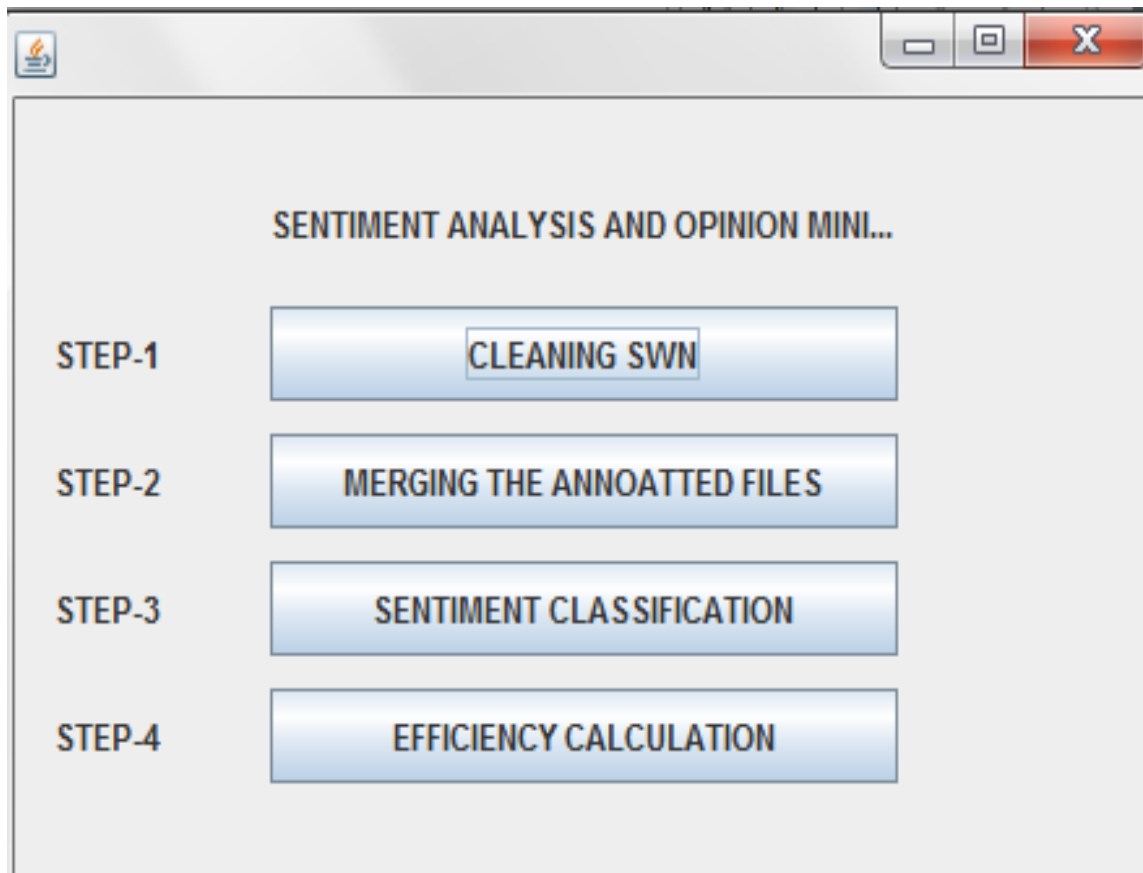
RECALL ACCURACIES
(Naive Bayes Revisited without Neutral class)

	MACHINE LEARNING	HYBRID
NAIVE BAYES	0.7015	0.7958

Fig. 4.22 NAÏVE BAYES WITHOUT NEUTRAL CLASS

5.OUTPUT SCREENS

Output File



Composition of Training Data

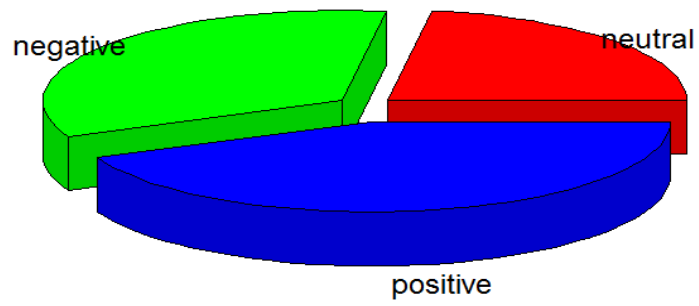


Fig.5.1 COMPOSITION OF DATASET

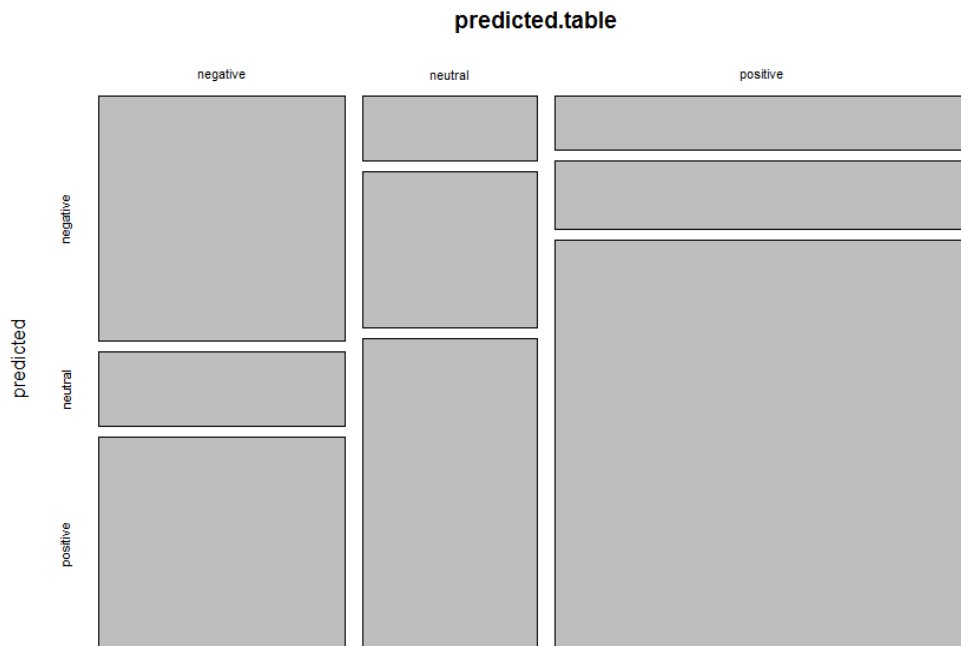


Fig.5.2 PLOT OF NAÏVE BAYES MACHINE LEARNING TECHNIQUE

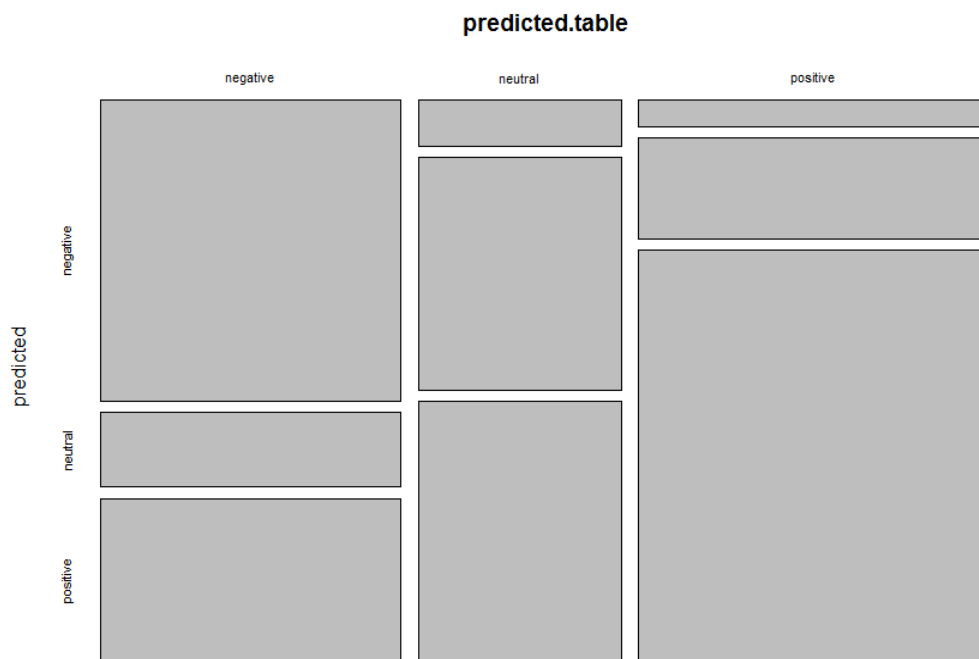


Fig.5.3PLOT OF NAÏVE BAYES HYBRID TECHNIQUE

RECALL ACCURACIES

	NAIVE BAYES	SUPPORT VECTOR MACHINE	RANDOM FOREST
MACHINE LEARNING	0.5790	0.6954	0.7279
HYBRID	0.6122	0.6974	0.7755

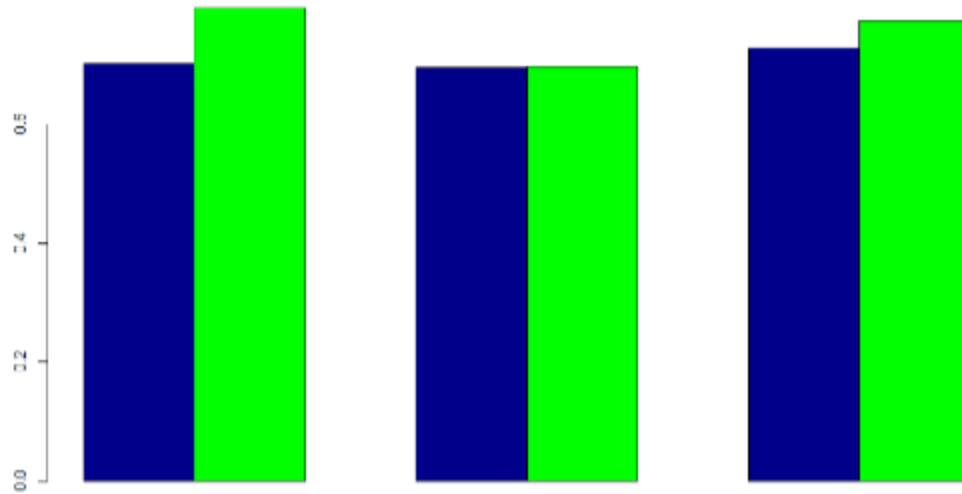


Fig. 5.4 COMPARING RECALL ACCURACIES OF NAÏVE BAYES , SVM & RANDOM FOREST WITH THEIR HYBRID ACCURACIES

(Dark blue- efficiency of machine learning techniques
Green – hybrid of their corresponding machine learning techniques)

RECALL ACCURACIES

(Naive Bayes Revisited without Neutral class)

	MACHINE LEARNING	HYBRID
NAIVE BAYES	0.7015	0.7958

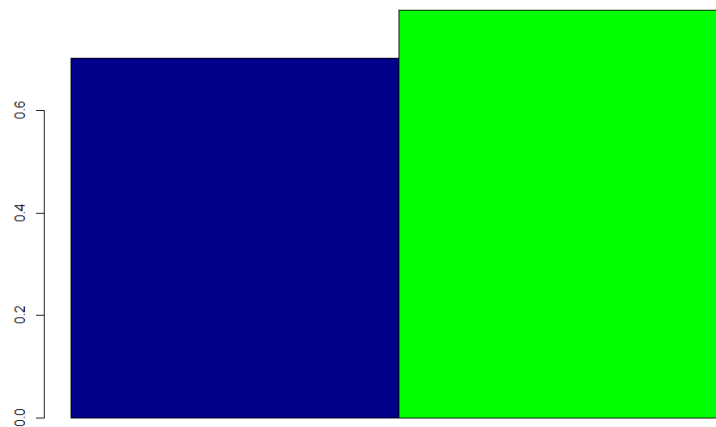


Fig 5.5 COMPARING RECALL ACCURACIES OF NAÏVE BAYES WITHOUT NUETRAL CLASS.

6.CONCLUSION

This research worked on both opinion lexicons (used SentiWordNet opinion Lexicon) and machine learning approaches(Naive Bayes , SVM, Random forest techniques) and devised a new hybrid technique combining the features of two, which shows improvements over all existing techniques. A further research was done using Naive Bayes technique by removing neutral reviews, and the accuracy was found to be increased in comparison to the research having all positive, negative and neutral reviews.

FUTURE SCOPE

Future aspects of our research will evolve around making or modifying our own dictionary resource that can be advantageous in overcoming some of the limitations seen in the context and also improving the efficiency by devising an algorithm to increase scores of hotel based feature words in the SentiWordNet dictionary.

The efficiency of various techniques with neutral reviews can be worked upon and improved . Web application for classification of hotel reviews will serve as a good way to commercialize and practically use this research in the current world.

We shall work upon building a fully fledged application that will provide an intuitive interface for the user to interact with and present the results of analysis in an interactive manner with some options to alter and focus on the more relevant part of the results.

7.REFERENCES AND BIBLIOGRAPHY

- [1] Esuli A, Sebastiani F., "SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining", Proceedings from International Conference on Language Resources and Evaluation (LREC), Genoa, 2006
- [2] S. Baccianella, A. Esuli, and F. Sebastiani, "Multi-facet rating of product reviews," Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, ser. ECIR '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 461–472.
- [3] U. Waltinger, "Germanpolarityclues: "A lexical resource for german sentiment analysis," Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC), 2010.
- [4] Hatem Ghorbel and David Jacot. 2011," Further experiments in sentiment analysis of french movie reviews", In Advances in Intelligent Web Mastering--3, pages 19--28. Springer: Advances in Distributed Agent-Based Retrieval Tools, pp.97-108
- [5] B. Pang and L. Lee - Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval, vol. 2, no. 1-2 (2008) 1-135
- [6] Hongning Wang, Yue Lu & ChengXiang Zhai: "Latent Aspect Rating Analysis without Aspect Keyword Supervision.", The 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'2011), P618-626, 2011.
- [7] Hongning Wang, Yue Lu and Chengxiang Zhai:" Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach.", The 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'2010), p783-792, 2010

- [8] Pranali Tumsare, Ashish .S. Sambare and Sachin .R. Jain, "Opinion Mining In Natural Language Processing Using Sentiwordnet and Fuzzy", June 2014.
- [9] Ohana, B. & Tierney, B., "Sentiment Classification of reviews using SentiWordNet", 9th.IT&T conference, Dublin, 2009
- [10] N. Godbole, M. Srinivasaiah, and S. Skiena: "Large-scale sentiment analysis for news and blogs", 2007
- [11] K. Yessenov, S. Misailović: "Sentiment Analysis of Movie Review Comments ", Spring, 2009
- [12] Baccianella S., Esuli A., & Sebastiani F., "SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining", Italy
- [13] K. Dave, S. Lawrence, and D. M. Pennock: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews (2003) 519-528.
- [14] M. Bautin, L. Vijayarenu & S. Skiena, "International Sentiment analysis for News and Blogs", NY
- [15] Akshay Amolik, Niketan Jivane, Mahavir Bhandari Dr.M.Venkatesan "Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques." International Journal of Engineering and Technology (IJET), Vol 7 No 6 Dec 2015-Jan 2016
- [16] W. Kasper, M. Vela, "Sentiment Analysis for Hotel Reviews", DFKI GmbH, 2011
- [17] S. Cerini, V. Compagnoni, A. Demontis, M. Formentelli, and G. Gandini. 2007. Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. In Andrea Sanjo, editor, Language resources and

linguistic theory: Typology, second language acquisition, English linguistics, pages 200–210. Franco Angeli Editore, Milano, IT.

[18] <http://sentiwordnet.isti.cnr.it/> accessed on August 10, 2016

[19] <http://www.cs.cmu.edu/~jiweil/html/hotel-review.html> / accessed on August 14, 2016