CHAPTER I

INTRODUCTION

I. INTRODUCTION

Sentiment Analysis also known as Opinion Mining is a field within Natural Language Processing (NLP) that builds systems that try to identify and extract opinions within text. Usually, besides identifying the opinion, these systems extract attributes of the expression e.g.:

- *Polarity:* If the speaker expresses a positive or negative opinion.
- *Subject :* The thing that is being talked about,
- *Opinion holder:* The person, or entity that expresses the opinion.

Currently, sentiment analysis is a topic of great interest and development since it has many practical applications. Since publicly and privately available information over Internet is constantly growing, a large number of texts expressing opinions are available in review sites, forums, blogs, and social media.

With the help of sentiment analysis systems, this unstructured information could be automatically transformed into structured data of public opinions about products, services, brands, politics, or any topic that people can express opinions about. This data can be very useful for commercial applications like marketing analysis, public relations, product reviews, net promoter scoring, product feedback, and customer service.

1.1 Sentiment Analysis Algorithms

There are many methods and algorithms to implement sentiment analysis systems, which can be classified as:

- **Rule-based** systems that perform sentiment analysis based on a set of manually crafted rules.
- **Automatic** systems that rely on machine learning techniques to learn from data.
- **Hybrid** systems that combine both rule based and automatic approaches.

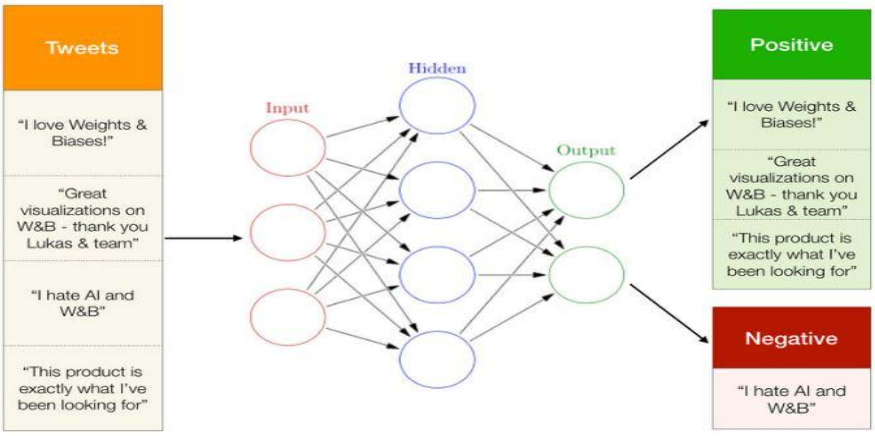
Tweets
"I love Weights & Biases!"
"Great visualizations on W&B - thank you Lukas & team"
"I hate AI and W&B"
"This product is exactly what I've been looking for"
Input
Hidden
Output
Positive
"I love Weights & Biases!"
"Great visualizations on W&B - thank you Lukas & team"
"This product is exactly what I've been looking for"
Negative
"I hate AI and W&B"

Figure 1. Processing in Sentiment analysis

1.2 Types of sentiment analysis

1. **Fine-grained sentiment analysis** provides a more precise level of polarity by breaking it down into further categories, usually very positive to very negative. This can be considered the opinion equivalent of ratings on a 5-star scale. e.g.: Very Positive = 5 stars and Very Negative = 1 star.
2. **Emotion detection** identifies specific emotions rather than positivity and negativity. Examples could include happiness, frustration, shock, anger and sadness.
3. **Intent-based analysis** recognizes actions behind a text in addition to opinion. For example, an online comment expressing frustration about changing a battery could prompt customer service to reach out to resolve that specific issue.eg:"Your customer support is a disaster. I've been on hold for 20 minutes".
4. **Aspect-based analysis** gathers the specific component being positively or negatively mentioned. For example, a customer might leave a review on a product saying the battery life was too short. Then, the system will return that the negative sentiment is not about the product as a whole, but about the battery life.

CHAPTER II

LITERATURE REVIEW

II. LITERATURE REVIEW

Sentiment analysis played a dominant role in the area of researches done by many researchers; there are many methods to carry out sentiment analysis. Many researches are going on to find out better alternatives due to its importance.

In [1], Soudamini Hota & Sudhir Pathak compares the K-Nearest neighbour [KNN] algorithm & Support Vector Machine[SVM] . It shows that the analysis is improved further by using KNN algorithm to train the classifier than the SVM technology. It is improved further by employing distance weighted KNN algorithm that involves associating weights with the nearest neighbors based on their proximity to the data point.

In [2], Lopamudra Dey compares the KNN algorithm with the Naive Bayes Classification. Their experimental results show that the classifiers yielded better results for the movie reviews with the Naïve Bayes‘approach. It giving above 80% accuracies and outperforming than the K-NN approach.

In [3], Surya Prakash Sharma says that, first extracts the feature, modifier and opinion from the dataset and then using clustering mechanism divide them into discrete clusters by user’s opinion. A feature wise opinion mining system to determine the polarity of the opinions in reviews documents using Senti-WordNet.

In [4], Devika M D had made a comparative study of different approaches in sentiment analysis. She conclude that, in the internet world majority of people depends on social networking sites to get their valued information, analyse he reviews from these blogs will yield a better understanding and help in their decision-making.

In [5], Wararat Songpan uses two methods to calculate the sentiment analysis called the Naive Bayes classification & Decision tree algorithm. The classifier model has calculated probability that shows value of trend to give the rating using naive bayes techniques, which gives correct classifier to 94.37° ~ compared with decision tree Techniques.

In [6], Vidisha M. Pradhan had done a Survey on Sentiment Analysis Algorithms for Opinion Mining. Dictionary based technique takes less processing time even though the accuracy is not up to the mark. However, the supervised learning approach provides better accuracy. From the survey, it summarizes the supervised techniques provide better accuracy compared to dictionary based approach.

In [7], Divyashree N explains the Opinion Mining and Sentimental Analysis of TripAdvisor.in for Hotel Reviews. The J48 algorithm is to classify Positive words and Negative words in reach review of all the hotels on the TripAdvisor website.

In [8], Smita Bhanap explains Sentiment analysis of mobile datasets using naïve bayes algorithm**.**Naïve Bayes algorithm is applied to train and test data to find sentiment polarity of overall sentence. In their paper,they present a lexicon-based approach for sentiment analysis on Twitter.They have used Naïve Bayes algorithm to find sentiment polarities of words in tweeter datasets of some mobile brands & evaluate the approach on various mobile brands datasets resulting into accuracy for sentiment polarity classification.

CHAPTER III

THEORETICAL BACKGROUND

3.1 THEORETICAL BACKGROUND

K-nearest neighbour classifier is one of the introductory supervised classifier, which every data science learner should be aware of. Fix &Hodges proposed K-nearest neighbour classifier algorithm in the year of 1951 for performing pattern classification task. For simplicity, this classifier is called as K-NN Classifier. To be surprised k-nearest neighbour classifier mostly represented as K-NN, even in many research papers too. Knn address the pattern recognition problems and also the best choices for addressing some of the classification related tasks. The simple version of the K-nearest neighbour classifier algorithms is to predict the target label by finding the nearest neighbour class. The closest class will be identified using the distance measures like Euclidean distance.

The KNN algorithm is a robust and versatile classifier that is often used as a benchmark for more complex classifiers such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM). Despite its simplicity, KNN can outer form more powerful classifiers and is used in a variety of applications such as economic forecasting, data compression and genetics. For example, KNN was leveraged in a2006 study of functional genomics for the assignment of genes based on their expression profiles.

3.2 CONCEPTS & METHODOLOGY

The main goal of the research is to analyse the data from the surveys and to decide whether it is suitable to be analysed with the use of the discussed data mining methods. The algorithm used here for classification are described below.

3.2.1 K-Nearest Neighbour Classifier

K-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. It is non parametric method used for classification or regression. In case of classification the output is class membership (the most prevalent cluster may be returned), the object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours. This rule simply retains the entire training set during learning and assigns to each query a class represented by the majority label of its k-nearest neighbours in the training set.

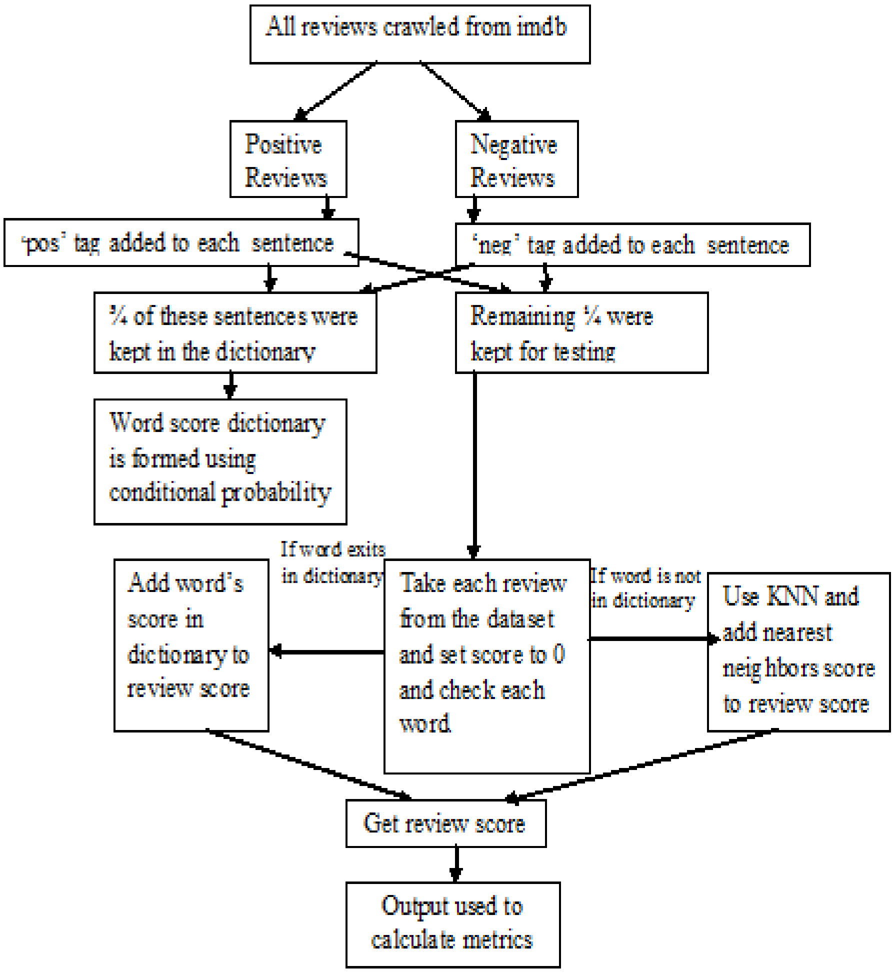
All reviews crawled from imdb
Positive Reviews
Negative Reviews
'pos' tag added to each sentence
'neg' tag added to each sentence
¾ of these sentences were kept in the dictionary
Remaining ¼ were kept for testing
Word score dictionary is formed using conditional probability
If word exits in dictionary
Add word's score in dictionary to review score
Take each review from the dataset and set score to 0 and check each word
If word is not in dictionary
Use KNN and add nearest neighbors score to review score
Get review score
Output used to calculate metrics

Figure 2. K-NN Classifier flowchart 1

The Nearest Neighbour rule (NN) is the simplest form of K-NN when K = 1.Given an unknown sample and a training set, all the distances between the unknown sample and all the samples in the training set can be computed. The distance with the smallest value corresponds to the sample in the training set closest to the unknown sample. Therefore, the unknown sample may be classified based on the classification of this nearest neighbour. The K-NN is an easy algorithm to understand and implement, and a powerful tool we have at our disposal for sentiment analysis. K-NN is powerful because it does not assume anything about the data, other than a distance measure can be calculated consistently between two instances. As such, it is called non-parametric or non-linear as it does not assume a functional form. The flowchart of k-nn classifier is given in Figure 3.2.1.

It is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbours. The case being assigned to the class is the most common among its K nearest neighbours measured by a distance function. These distance functions can be Euclidean, Manhattan, Minkowski, Levenshtein and Hamming distance. If K = 1, then the case is simply assigned to the class of its nearest neighbour. At times, choosing K turns out to be a challenge while performing KNN modeling. The algorithm looks at different centroids and compares distance using some sort of function (usually Euclidean), then analyzes those results and assigns each point to the group so that it is optimized to be placed with all the closest points to it.

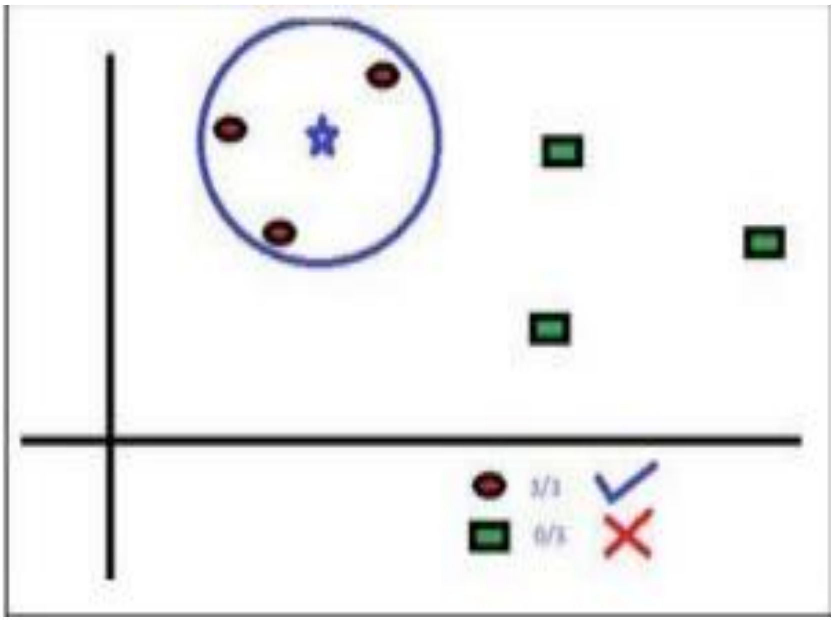

Figure 3. Example for KNN algorithm

The score can be calculated using:

Positivity Score = $\left(\sum_1^j \text{score (pos)} + \sum_1^k \text{score (neg)}\right) \Big/ \sum_1^s \text{maximum score}$

-Equation (3.2.1)

Here s=j+k, ie. Count of both positive and negative together. In weighted k-NN method they first of all tokenize the sentences and removed the stop words from the comments they have fetched. The algorithm proposed by the authors of is carried out in two parses. A positive score is assigned to each reviews after the first parse. This is passed for second parsing and an input of neutral review is given. Using this the score is modified if required. It is done for better positivity determination and an output file consisting of review ID and its positive score is determined.

Strings are broken into tokenized arrays of single words. These words are analysed against TXT files that contain emotion words with ratings, emoticons with ratings,booster words with ratings and possible polarity changers. A score is then calculated based on this analyse and this forms the "Sentiment analysis score".

3.2.2 Pseudo-code for KNN:

We can implement a KNN model by following the below steps:

1. Load the data
2. Initialise the value of k
3. For getting the predicted class, iterate from 1 to total number of training data points

1. Calculate the distance between test data and each row of training data.

   Here we will use Levenshtein Distance as our distance metric since it's the most popular method.

   The other metrics that can be used are Chebyshev,Levenshtein, cosine, etc.
2. Sort the calculated distances in ascending order based on distance values
3. Get top k rows from the sorted array
4. Get the most frequent class of these rows
5. Return the predicted class

3.2.3 Levenshtein Distance:

The Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions, or substitutions) required to change one word into the other**.**

$$\boldsymbol{lev_{a,b}(i,j)} = \begin{cases} \boldsymbol{max(i,j)} & \boldsymbol{if min(i,j) = 0,} \\ \boldsymbol{min} \begin{cases} \boldsymbol{lev_{a,b}(i-1,j)+1} \\ \boldsymbol{lev_{a,b}(i,j-1)+1} \\ \boldsymbol{lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)}} \end{cases} & \boldsymbol{otherwise} \end{cases}$$

-Equation(3.2.2)

Here, 1($a_i \neq b_i$) is the indicator function equal to 0 when $a_i \neq b_i$ and equal to 1 otherwise, and $\boldsymbol{lev_{a,b}(i,j)}$ is the distance between the first i characters of a and the first j characters of b.

3.2.4 Phrase Analysis

This function is key to identifying whether the phrase in questions can be compared to phrases that we have analysed and stored before. It uses Levenshtein distance to calculate distance between word length phrases against the dataset we already have. We also make use of PHP's similar_text to double verify proximity.This means that the more phrases we have analysed previously improves the entire dataset and allows phrases to be more accurately scored against historical data.

1. The phrase is broken up into n-gram lengths.
2. The array is reverse sorted so we compare 10 word length phrases first, then 9, and so on
3. Phrases are matched against positive, negative and neutral phrases in the relevant TXT files
4. Only matches that meet the minimum levenshtein min distance &similiarity_min_distance are kept.

3.3 DESIGN & MODELLING

3.3.1 Diagrams

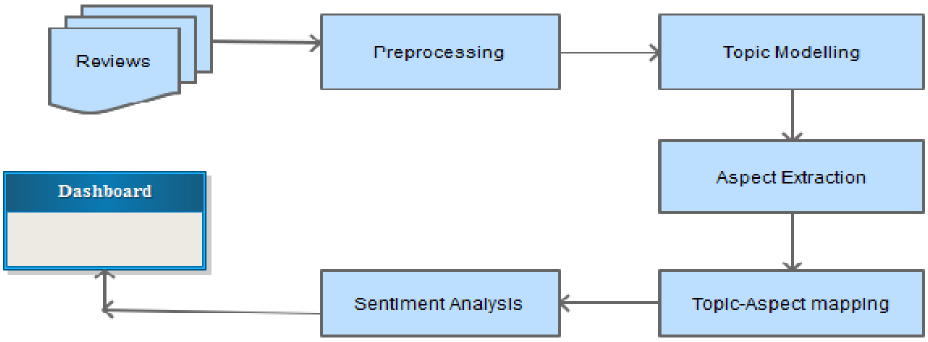

Figure 4. Opinion Mining / Sentiment Analysis Process

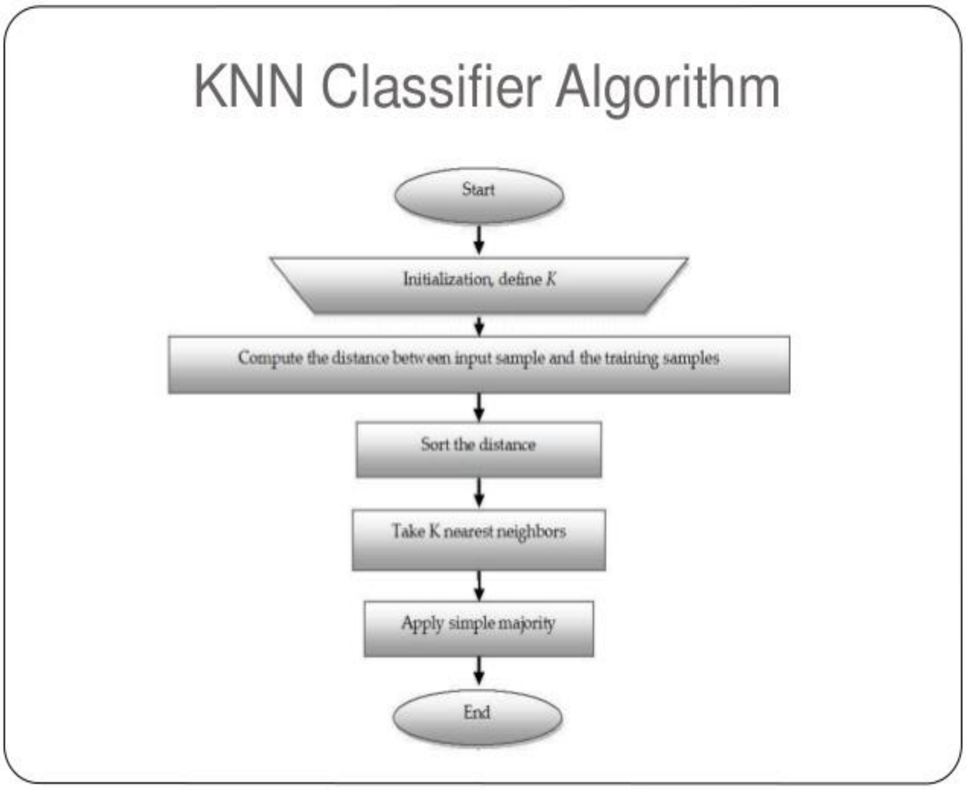

Figure 5. K-NN Classifier flowchart 2.

3.3.2 Table Design

1.tbl_userfeedback

Primary Key : uf_id

Foreign Key : r_id,h_id[references from tbl_registration & tbl_addresort]

Field Name	**Data Type**	**Length**	**Description**	**Contraint**
uf_id	INT	2	Rate_id	Primary Key
r_id	INT	2	User id	Foreign key from tbl_registration
h_id	INT	10	Resort id	Foreign key from tbl_addresort
uf_msg	Varchar	10	Review Message	Null
uf_date	Date	2	Date	Null
uf_rate	INT	10	Rating	Null
rate_status	INT	1	Status	Null

Table 1. tbl_userfeedback

2.tbl_registration

Primary Key : r_id

Field Name	**Data Type**	**Length**	**Description**	**Constraint**
r_id	int	2	User id	Primary Key
r_name	varchar	10	First name	Null
r_lname	varchar	10	Last name	Null
r_address	varchar	10	Address	Null
r_dob	date	5	Date of Birth	Null
r_gender	varchar	10	Gender	Null
r_phone	varchar	10	Phone number	Null
r_email	varchar	20	Email	Null

r_status	int	2	Status	Null

Table 2. tbl_registration

3.tbl_addresort

Primary Key : h_id

Foreign Key : r_id,d_id [references from tbl_registration & tbl_district]

Field Name	Data Type	Length	Description	Constraint
h_id	int	2	Resort id	Primary Key
r_id	int	2	User_id	Foreign key from tbl_registration
h_name	varchar	10	Resort Name	Null
h_address	varchar	10	Resort Address	Null
h_star	int	2	Resort Star classification	Null
d_id	int	2	District id	Foreign key from tbl_district
s_id	int	2	State	Null
h_pin	int	10	Pin number	Null
h_location	varchar	10	Resort location	Null
h_des	varchar	20	Resort Description	Null
h_open	time	5	Opening time	Null
h_close	time	5	Closing Time	Null
h_phone	varchar	10	Phone number	Null
h_url	varchar	10	Site url	Null
h_email	varchar	10	Email	Null
h_image	varchar	20	Image	Null

Table 3. tbl_addresort

CHAPTER IV

PROJECT IMPLEMENTATION

IV. PROJECT IMPLEMENTATION

The opinion mining has become one of popular research area. The challenge is in process of opinion mining or sentiment analysis that is unstructured and noisy data on website.

There are mainly three types of opinion mining techniques (figure. 4.1)**:**

1. **Supervised Learning Techniques** : The most widely used supervised learning techniques are Support Vector Machines(SVM), Neural network, Multi-Layer Perceptron (MLP), Decision tree, Naïve Bayes(NB) Classification, Maximum Entropy(MaxEnt),K-NN Algorithm.
2. **Unsupervised Learning Techniques:** Mostly used technique are Clustering algorithm, expectation maximization algorithm, matrix factorization, principal component analysis.
3. **Case-Based Reasoning:** It is an emerging artificial techniques. CBR is an intelligent tool of computer reasoning and solves the problem in such a real time scenario. Solution is stored in CBR repository also known as case base.

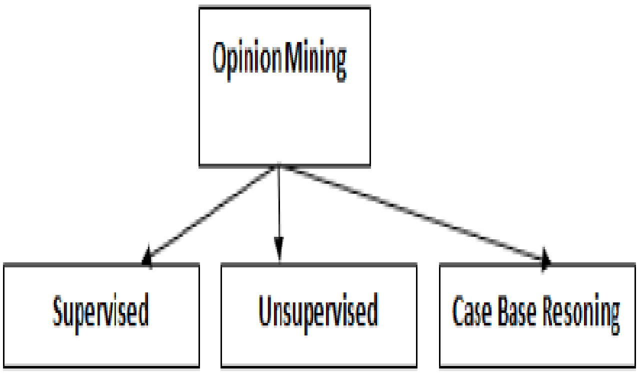

Figure 6. Types of Opinion Mining Techniques

A part of opinion mining refers using of natural language processing (NLP) by proposed different method of dictionary for sentiment analysis.They tried to extract word from sentences for removal stop word or unnecessary word automatically. In addition, various dictionaries are solved by machine learning methods which try to rank scoring of various dictionaries. For example, K-NN algorithm to collect the ranking of different dictionary into rule for classify the opinion.After word segmentation process is removal stop words by dictionary checking. It focuses on the calculating polarity of words to trend in positive , negative or nuetral in a cluster of interest's customer that are extracted from texts and compared the word occurrence of whole sentence. If the word extractions have weight from dictionary of emotional words, it is calculated to answer the comment as positive or negative.

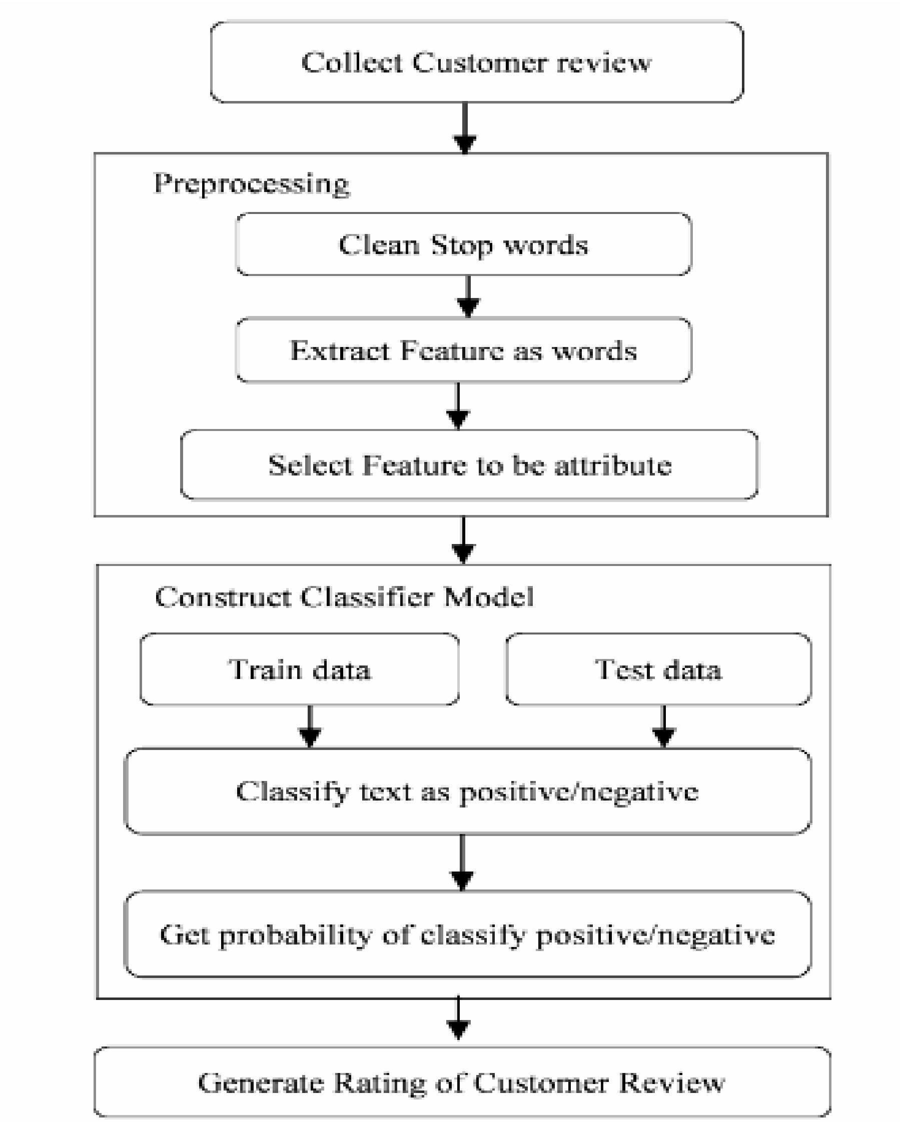
Collect Customer review
Preprocessing
Clean Stop words
Extract Feature as words
Select Feature to be attribute
Construct Classifier Model
Train data
Test data
Classify text as positive/negative
Get probability of classify positive/negative
Generate Rating of Customer Review

Figure 7. Proposed Methodology for generating score of customer review using opinion mining

The proposed methodology used customer review's resorts from a website which service in resort reservation directly. The target of classify customer review from this website because the comment is posted from customer who is serviced checked-in and checked-out from resort. The system has cleaned the promotion of resort's comment which has only existed customer review given comment.The open opinion texts are collected customer reviews that are used service to checked-in/out the resorts. The process is started from collected data and preprocessing is cleaned data by removal stop words,Calculate the distance between test data and each row of training data. Using Levenshtein Distance sort the calculated distances in ascending order based on distance values.Get top k rows from the sorted array and the most frequent class of these rows and return the predicted class.

CHAPTER V

RESULTS

V. RESULTS

Opinion Mining for Resort Review system is a natural language processing system that detects hidden sentiments in reviews of the customer and rates the reviews accordingly.

The precision of the proposed system is approx. 82 percent. The recall of the proposed system is 81.5 percent. The accuracy achieved by the proposed system is up to 86 percent.

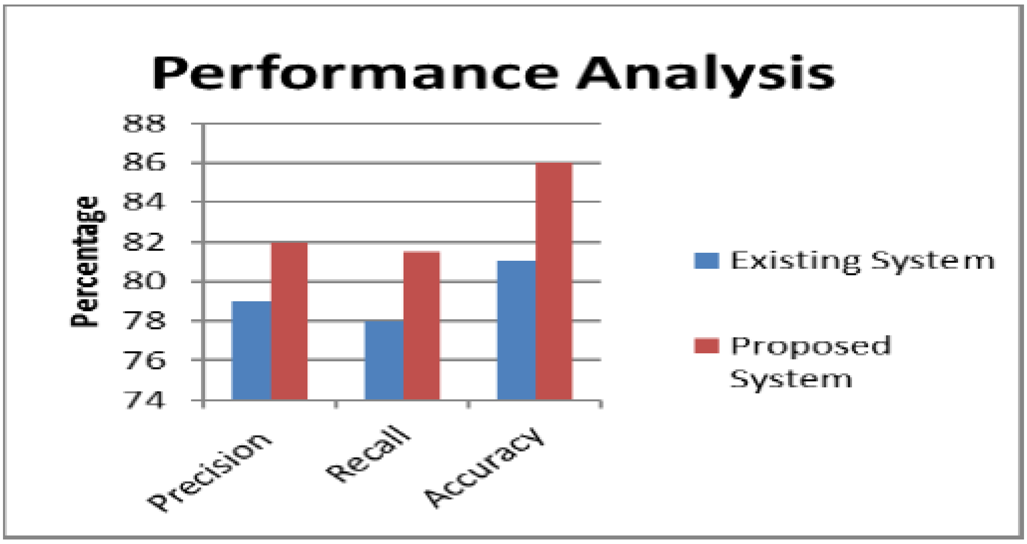

Figure 8. Performance Analysis

When processing time is considered it is shown that the processing time is totally depend upon the size of test set as the size increases the processing time increases and remain same for these classifiers and if different number of documents (and of different test size) are used then we can observed the processing time differences.

CHAPTER VI

CONCLUSIONS AND SCOPE FOR FUTURE WORK

VI. CONCLUSIONS AND SCOPE FOR FUTURE WORK

Various sentiment analysis methods and its different levels of analyzing sentiments have been studied here. Our ultimate aim is to come up with Sentiment Analysis which will efficiently categorize various reviews. Research work is carried out for better analysis methods in this area, We use the K-Nearest Neighbour algorithm to effectively calculate the polarity of the reviews. In the world of Internet majority of people depend on social networking sites to get their valued information, analysing the reviews from these blogs will yield a better understanding and help in their decision-making.

Sentiment Analysis Scope:

Sentiment analysis can be applied at different levels of scope:

- Document level sentiment analysis obtains the sentiment of a complete document or paragraph.
- Sentence level sentiment analysis obtains the sentiment of a single sentence.
- Sub-sentence level sentiment analysis obtains the sentiment of sub-expressions within a sentence.

The research scope in opinion mining and sentiment analysis are:

- Spam Detection using Sentiment Analysis.
- Sentiment Analysis on short Sentence that include abbreviations.
- Improvement of existing sentiment word identification algorithm.
- Developing fully automatic tools for analysis.
- Effective Analysis of policy documents which containing opinion content.
- Managing the of bi polar sentiments successfully.
- Designing and Generation of highly content corpus database.

Future Work

Opinion mining and sentimental analysis is an emerging field of data mining used to extract the knowledge from a huge volume of customer comments, feedback and reviews on any product or topic etc. A lot of work in opinion mining in customer reviews has been conducted to mine opinions in form of document, sentence and feature level sentiment analysis. In future, Opinion Mining can be carried out on set of discovered feature expressions extracted from reviews. The Opinion Mining and in natural language processing community, Sentiment Analysis become a most interesting research area. A more innovative and effective techniques needed to be invented which should overcome the current challenges faced by Opinion Mining and Sentiment Analysis.

APPENDIX 1

Sample Project code/ Algorithm

```
<?php
include ('Research/lib/sentiment_analyser.class.php');
$sa = new SentimentAnalysis();
$sa->initialize();
if (isset($_POST['save_dats'])) {
$rating = $_POST['rating'];
$text = $_POST['text'];
$sa->import_sentiment_custom($text,$rating);
die();
} else { ?><!doctype html>
<body><div class='wrapper'>
<?php
if (isset($_POST['sent_data'])){ ?>
<div class='returned_data'>
<?php
$sent_data = explode("\n",$_POST['sent_data']);
$min_submit_lev_score = $sa->return_levenshtein_min_submit_distance();
$analysed_array = array();
$i = 0;
foreach ($sent_data as $dataset) {
$original_data = $dataset;
$check = $sa->analyse($dataset);
$rating = $sa->return_sentiment_rating();
$ratings_data = $sa->return_sentiment_calculations();

//echo $ratings_data;

$analysed_array[$i]['dataset'] = implode(' ',$sa->return_tokenized_mention());

$analysed_array[$i]['original_dataset'] = $original_data;

$analysed_array[$i]['rating'] = $rating;

$analysed_array[$i]['preferred_match_type'] = $sa->return_preferred_match_type();

$analysed_array[$i]['sentiment_analysis'] = $sa->return_sentiment_analysis();

$analysed_array[$i]['proximity_analysis'] = $sa->return_phrase_proximity();

$i++;}

foreach($analysed_array as $key => $output) {

$allow_submission = false;

//var_dump($output);

if ($output['preferred_match_type'] == 'sentiment_analysis' ||
$output['proximity_analysis'][1]['levenshtein'] > $min_submit_lev_score) {

if (count(explode(" ",$output['dataset'])) < 4) {

$allow_submission = false;

} else {$allow_submission = true;}

} else {

$allow_submission = false;}

echo "<tr id='tr_".$key."'>";

if ($allow_submission) {}

else{

echo "<td>&nbsp;</td>";}}

include 'dbconnect.php';
```

```
$data=$output['dataset'];

$o=$output['rating'];

$d=date("Y-m-d");

$a=$_SESSION['r_id'];

$sql=mysqli_query($con,"SELECT * FROM `tbl_addresort` WHERE `h_id`='$kid'");

$sql=mysqli_query($con,"SELECT * FROM `tbl_registration` WHERE `r_id`='$a'");

$sql=mysqli_query($con,"INSERT INTO `tbl_userfeedback`(`r_id`, `h_id`, `uf_msg`, `uf_date`,
`uf_rate`, `uf_status`) VALUES ('$a','$kid','$data','$d','$o','1')");

echo "</div>";

?>

<?php} ? ?>

</form><div class="reviews_title">reviews</div>

<?php

include 'Rating.php';

$rating = new Rating();

$itemList = $rating->getItemList();

foreach($itemList as $item){

$average = $rating->getRatingAverage($item["h_id"]);

$itemDetails = $rating->getItem($_GET['uid']);

?>

<?php } ?>

<?php

$itemRating = $rating->getItemRating($_GET['uid']);

$ratingNumber = 0;
```

```
$count = 0;

$fiveStarRating = 0;

$fourStarRating = 0;

$threeStarRating = 0;

$twoStarRating = 0;

$oneStarRating = 0;

foreach($itemRating as $rate){

$ratingNumber+= $rate['uf_rate'];

$count += 1;

if($rate['uf_rate'] >4.5 and  $rate['uf_rate'] <=5) {

$fiveStarRating +=1;

} else if($rate['uf_rate'] >3.5 and  $rate['uf_rate'] <=4.5) {

$fourStarRating +=1;

} else if($rate['uf_rate']> 2.5 and  $rate['uf_rate'] <=3.5) {

$threeStarRating +=1;

} else if($rate['uf_rate'] >1.5 and  $rate['uf_rate'] <=2.5) {

$twoStarRating +=1;

} else if($rate['uf_rate'] >0 and  $rate['uf_rate'] <=1) {

$oneStarRating +=1;}}

$average = 0;

if($ratingNumber&& $count) {

$average = $ratingNumber/$count;}

?>

<h4>Rating and Reviews</h4>
```

```
<h2 class="bold padding-bottom-7"><?phpprintf('%.1f', $average); ?><small>/ 5</small></h2>
<?php
$averageRating = round($average, 0);
for ($i = 1; $i <= 5; $i++) {
$ratingClass = "btn-default btn-grey";
if($i <= $averageRating) {
$ratingClass = "btn-warning";}
?>
<button type="button" class="btnbtn-sm<?php echo $ratingClass; ?>" aria-label="Left Align">
<span class="glyphiconglyphicon-star" aria-hidden="true"></span>
</button>
<?php } ?>
<?php
$fiveStarRatingPercent = round(($fiveStarRating/5)*100);
$fiveStarRatingPercent= !empty($fiveStarRatingPercent)?$fiveStarRatingPercent.'%':'0%';
$fourStarRatingPercent = round(($fourStarRating/5)*100);
$fourStarRatingPercent= !empty($fourStarRatingPercent)?$fourStarRatingPercent.'%':'0%';
$threeStarRatingPercent = round(($threeStarRating/5)*100);
$threeStarRatingPercent= !empty($threeStarRatingPercent)?$threeStarRatingPercent.'%':'0%';
$twoStarRatingPercent = round(($twoStarRating/5)*100);
$twoStarRatingPercent= !empty($twoStarRatingPercent)?$twoStarRatingPercent.'%':'0%';
$oneStarRatingPercent = round(($oneStarRating/5)*100);
$oneStarRatingPercent= !empty($oneStarRatingPercent)?$oneStarRatingPercent.'%':'0%';
?>
```

```
<div class="progress-bar progress-bar-success" role="progressbar" aria-valuenow="5" aria-valuemin="0" aria-valuemax="5" style="width: <?php echo $fiveStarRatingPercent; ?>">

<span class="sr-only"><?php echo $fiveStarRatingPercent; ?></span></div>

<div class="pull-right" style="margin-left:10px;"><?php echo $fiveStarRating; ?></div></div>

<div class="progress-bar progress-bar-primary" role="progressbar" aria-valuenow="4" aria-valuemin="0" aria-valuemax="5" style="width: <?php echo $fourStarRatingPercent; ?>">

<span class="sr-only"><?php echo $fourStarRatingPercent; ?></span></div>

<div class="pull-right" style="margin-left:10px;"><?php echo $fourStarRating; ?></div>

<div class="progress-bar progress-bar-info" role="progressbar" aria-valuenow="3" aria-valuemin="0" aria-valuemax="5" style="width: <?php echo $threeStarRatingPercent; ?>">

<span class="sr-only"><?php echo $threeStarRatingPercent; ?></span></div>

<div class="pull-right" style="margin-left:10px;"><?php echo $threeStarRating; ?></div>

<div class="progress-bar progress-bar-warning" role="progressbar" aria-valuenow="2" aria-valuemin="0" aria-valuemax="5" style="width: <?php echo $twoStarRatingPercent; ?>">

<span class="sr-only"><?php echo $twoStarRatingPercent; ?></span></div>

<div class="pull-right" style="margin-left:10px;"><?php echo $twoStarRating; ?></div>

<div class="progress-bar progress-bar-danger" role="progressbar" aria-valuenow="1" aria-valuemin="0" aria-valuemax="5" style="width: <?php echo $oneStarRatingPercent; ?>">

<span class="sr-only"><?php echo $oneStarRatingPercent; ?></span></div>

<div class="pull-right" style="margin-left:10px;"><?php echo $oneStarRating; ?></div>

<?php

include 'dbconnect.php';

$kid1=$_SESSION['r_id'];

$res1=mysqli_query($con,"SELECT * FROM `tbl_userfeedback` f1 LEFT JOIN `tbl_registration` r1 on f1.r_id=r1.r_id where f1.h_id='$kid'");

while($row1=mysqli_fetch_array($res1)){
```

```
?>
<div class="review_image">
<imgsrc="Uploads/<?php echo $row1['r_image'];?>" alt="https://unsplash.com/@saaout"></div>
<div class="review_title"><?php echo $row1['r_name'];?></div>
<div class="review_rating"><?php echo $row1['uf_rate'];?></div>
<div><span id="stars<?php echo $row1['uf_id'];?>"></span>
<script type="text/javascript">
document.getElementById("stars<?php echo $row1['uf_id'];?>").innerHTML = getStars(<?php echo $row1['uf_rate'];?>);
functiongetStars(rating) {
rating = Math.round(rating * 2) / 2;
let output = [];
for (var i = rating; i >= 1; i--)
output.push('<i class="fafa-star" aria-hidden="true" style="color: gold;"></i>&nbsp;');
if (i == .5) output.push('<i class="fafa-star-half-o" aria-hidden="true" style="color: gold;"></i>&nbsp;');
for (let i = (5 - rating); i >= 1; i--)
output.push('<i class="fafa-star-o" aria-hidden="true" style="color: gold;"></i>&nbsp;');
returnoutput.join(");}
</script></div><div class="review_text"><p><?php echo $row1['uf_msg'];?></p></div>
<div class="review_date"><?php echo $row1['uf_date'];?></div>
<?php} }?>
```

REFERENCES

References

[1] Soudamini Hota , Sudhir Pathak, “KNN classifier based approach for multi-class sentiment analysis of twitter data”- *International Journal of Engineering & Technology, 7 (3) (2018) 1372-1375*

[2] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, “Sentiment Analysis of Review Datasets Using Naïve Bayes‘ and K-NN Classifier,” *I.J. Information Engineering and Electronic Business,* 2016.

[3] Surya Prakash Sharma, Dr Rajdev Tiwari, Dr Rajesh Prasad ,“Opinion Mining and Sentiment Analysis on Customer Review Documents”- A Survey, *International Conference on Advances in Computational Techniques and Research Practices-Vol. 6, Special Issue 2, February 2017*

[4] Devika M D, Sunitha C, Amal Ganesha, “Sentiment Analysis:A Comparative Study On Different Approaches”- *Procedia Computer Science 87 (2016).*

[5] Wararat Songpan ,*“*The Analysis and Prediction of Customer Review Rating Using Opinion Mining”-, *2017 IEEE SERA 2017, June 7-9,2017, London, UK.*

[6] Vidisha M. Pradhan, Jay Vala,Prem Balani ,“A Survey on Sentiment Analysis Algorithms for Opinion Mining”-, *International Journal of Computer Applications (0975 – 8887) Volume 133 – No.9, January 2016*

[7] Amit G. Shirbhate, Sachin N. Deshmukh ,“Feature Extraction for Sentiment Classification on Twitter Data”-, *International Journal of Science and Research (IJSR), Volume 5 Issue 2, February 2016*

[8] Smita Bhanap, Dr. Seema Kawthekar ,“Sentiment analysis of mobile datasets using naïve bayes algorithm”-, *Volume 9, No. 2, March-April 2018 International Journal of Advanced Research in Computer Science.*