**Project Status Report**

**B.Tech Computer Science and Engineering**

**Motilal Nehru National Institute of Technology, Allahabad**

1. GROUP NO (If ANY): CS-36

2. Department/Program: Computer Science and Engineering (CSE)

3. Title of the Project: Movie Review Sentiment Analysis

4. Mentor Name: Prof. R. S. Yadav (Faculty CSED, MNNIT Allahabad)

5. Status of the Project: Completed (Project files are also included)

6. Origin of the Project:

According to Ramteke et al. (2012) motivation for Sentiment Analysis is two-fold. Both consumers and producers highly value “customer’s opinion” about products and services. Thus, Sentiment Analysis has seen a considerable effort from industry as well as academia.

The Consumer’s Perspective:

While taking a decision it is very important for us to know the opinion of the people around us. Earlier

this group used to be small, with a few trusted friends and family members. But, now with the advent

of Internet we see people expressing their opinions in blogs and forums. These are now actively read by

people who seek an opinion about a particular entity (product, movie etc.). Thus, there is a plethora of

opinions available on the Internet.

From a consumers’ point of view extracting opinions about a particular entity is very important. Trying

to go through such a vast amount of information to understand the general opinion is impossible for

users just by the sheer volume of this data. Hence, the need of a system that differentiates between

good reviews and bad reviews. Further, labeling these documents with their sentiment would provide a

succinct summary to the readers about the general opinion regarding an entity.

The Producer’s Perspective

With the explosion of Web 2.0 platforms such as blogs, discussion forums, etc., consumers have at their

disposal, a platform to share their brand experiences and opinions, positive or negative regarding any

product or service. According to Pang and Lee (2008) these consumer voices can wield enormous influence

in shaping the opinions of other consumers and, ultimately, their brand loyalties, their purchase decisions,

and their own brand advocacy.

Since the consumers have started using the power of the Internet to expand their horizons, there has

been a surge of review sites and blogs, where users can perceive a product’s or service’s advantages and

faults. These opinions thus shape the future of the product or the service. The vendors need a system

that can identify trends in customer reviews and use them to improve their product or service and also

identify the requirements of the future.

The Societies’ Perspective

Recently, certain events, which affected Government, have been triggered using the Internet. The social

networks are being used to bring together people so as to organize mass gatherings and oppose oppression.

On the darker side, the social networks are being used to insinuate people against an ethnic group or

class of people, which has resulted in a serious loss of life. Thus, there is a need for Sentiment Analysis

systems that can identify such phenomena and curtail them if needed.

8. Importance of the proposed project in the context of current status and its relevance to computer science and engineering

**Use case: How KFC is doing it?**

An excellent example of how to use sentiment analysis for brand building and monitoring is KFC. For a while, KFC was stuck in the past, while the competition was moving ahead and reinventing themselves with the narratives of healthy food and feel-good experiences.

So, instead of trying to establish themselves in the crowded niche, KFC had chosen to use the ubiquitous power of the brand. KFC started riding on the wave’s memes and pop culture iconography (most recently by using RoboCop to promote the newest product) to instil the brand’s value proposition.

This approach generates natural traction around the brand that is augmented by the pop culture reference. As a result, users engage with the brand and ultimately are led to engage with the product down the line. You have to react and adapt almost instantly, which is where sentiment analysis kicks in.

The method combines sentiment analysis in social networks monitoring and campaign management that involves:

* Performance monitoring with aspect-based sentiment analysis to point out the specific elements of the presentation
* Topic mining to extract new ideas and variations.

This creates a loop which perpetuates the campaign’s proceedings.

Due to the nature of the marketing campaign, the users are actively involved in commenting or reacting to the ad content. In turn, this generates further ideas for the development of the campaign.

The result:

* KFC brand is constantly present in the media landscape and that presence guarantees the steady growth of the reach and ultimately the market share.

#### Example: How Apple is doing it

The way Apple presents its products and establishes them on the market is a fine example of sentiment analysis application for the benefit of market research and competitor analysis.

Think about how neatly the product’s strong points fit general pains and disgruntlement of the various segments of the user.

For example:

* Bad design - you don’t even need to think when using our stuff.
* Poor privacy - we keep personal data use at an absolute minimum.
* Low battery life - we’ve got resource management tools.

Such things can be pointed out by analysing the competitors and their movements on the market in general by specific aspects. For example:

* Brand value proposition
* Addressing various issues
* Introducing new features
* Announcing milestones and so on

A combination of this information from several maps out the market situation and allows calculating an additional perspective on how to differentiate and strengthen its value proposition.

The result:

* Apple is a trillion-dollar company because they listen to the customer.

### **Example: Product Analytics**

The use of sentiment analysis in product analytics stems from reputation management. Conceptually, it is very similar to brand monitoring. But instead of brand mentions, it goes for specific comments and remarks regarding the product and its performance in specific areas (user interface, feature performance, etc).

This kind of insight is very important at the initial stages with MVP when you need to try the product by fire (i.e. actual users) and make it as polished as possible.

At this stage, the most basic way to apply sentiment analysis is to gather and categorize feedback for further improvements.

Sentiment analysis algorithm can do the dirty work and show what kind of feedback goes from which segment of the audience and at what it points.

Usually, the whole thing is divided between the following types:

* Brand keywords
* Brand-adjacent keywords
* Customer needs
* Customer sentiment
* Competitors analysis (based on similar criteria)

As a result, this can be a significant factor in the product’s successful establishment on the market.

At the later stages, the use of sentiment analysis in product analytics merges with brand monitoring and provides a multi-dimensional view on the product and its brand:

* How the brand/product is perceived by various target audience segments?
* Which elements of the product or its presentation are the points of contention and in what light?

#### Use case: How Google is doing it?

A good showcase of how sentiment analysis application contributes to product improvement can be seen in Google’s output. Let’s take Chrome browser for example.

Google Chrome’s development team is constantly monitoring user feedback, whether it is direct or indirect (i.e. presented in the open sources, most notably, blogs).

But they are not looking at feedback as a message from the user but rather as a sum of its parts:

* the sentiment itself (positive or negative)
* Mentions of the specific aspects of the product - whether it is scalability, extensions, security or UI.
* Sentiments, wishes, and recommendations regarding the product in general and its specific elements.

The result:

* These elements provide an additional perspective on the weak and strong points of the product
* This subsequently contributes to further research and development of the product

9. Work Plan

**Problem Scenario**

We need to train a model to recognise the emotion of the movie review as positive or negative. It is a binary classification problem.

The dataset used here is known as 'Polarity Dataset'. The dataset used in our project has been taken from Stanford .

This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets.

This dataset has following properties:

* They provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing.
* It only comprises of English reviews.
* There is whitespace around punctuations like periods, commas, and brackets.
* There is additional unlabelled data for use as well.

1. Methodology:

This problem is approached in three stages:

**A. Data Pre-processing**

 This stage involves the following steps:

1. Loading the data
2. Analysis of data
3. Cleaning the data.
4. Defining a vocabulary
5. Creating the feature list (words that would be considered in the   sentiment analysis)

**B. Bag of Words Representation**

This stage focuses on preparing the data for the training model. It      involves following steps:

1. Converting reviews to lines of tokens.
2. Encoding reviews with a bag-of-words model representation.

**C. Sentiment Analysis Model**

In this project we develop a **Naive Bayes Model** to predict the sentiment/emotion of the reviews.

1. Time Schedule of activities

**A. Data Pre-processing**

**A.1. Training and Testing set**

The dataset contains positive and negative reviews. Among them, some positive and negative reviews are used as training sets and the remaining positive and negative reviews are used as test data.

## A.2. Load the data

The **load** method helps to load the dataset into the memory.

## A.3. Analysis of the data

In the analysis of the data we generate a report on:

* Size of the dataset
* Total Positive Sentiment in the dataset
* Total Negatives Sentiment in the dataset
* Average Positive review length
* Average Negative review length
* Train Data size
* Test Data size

# 

# **B. BAG OF WORDS REPRESENTATION**

## **B.1. Define a Vocabulary**

1. **Cleaning the data:**

This step helps to turn data into clean tokens. The following methods are employed in this project to generate clean tokens.

* Remove all the punctuations.
* Remove all stop words.
* Remove all words with small length (length <= 1 character).

**What are Stopwords?**

**Stop words** are words which are filtered out before or after processing o[f](https://en.wikipedia.org/wiki/Natural_language_processing) natural language data. Though "stop words" usually refers to the most common words in a language, there is no single universal list of stop words used by all natural language processing tools, and indeed not all tools even use such a list.

Stopwords Example:

'a', 'about', 'above', 'across', 'after', 'afterwards', 'again', 'against', 'all', 'almost', 'alone','along', 'already', 'also', 'although', 'always', 'am', 'among', 'amongst', 'amongst', 'amount', 'an', 'and', 'another', 'any', 'anyhow', 'anyone', 'anything', 'anyway', 'anywhere', 'are', 'around','as', 'at', 'back', 'be', 'became', 'because', 'become', 'becomes', 'becoming', 'been', 'before', 'beforehand', 'behind', 'being', 'below', 'beside', 'besides', 'between', 'beyond', 'bill', 'both', 'bottom', 'but', 'by', 'call', 'can', 'cannot', 'cant', 'co', 'con', 'could', 'couldn't', 'cry', 'de',  'describe', 'detail', 'did', 'do', 'does', 'doing', 'don', 'done','down', 'due', 'during', 'each', 'eg', 'eight', 'either', 'eleven', 'else' …….(and many more)

1. **Build Vocabulary**

It is necessary to define a vocabulary of words to vectorise the document of reviews. The below method defines a vocabulary of words and maintains the count of the occurrence of each word in the vocabulary.

The **cleanbuildVocabulary** functionof the **Features** class performs cleaning and building the vocabulary

1. **Build Features**

From the vocabulary created above features are created on the basis

          of these features our reviews are classified as Positive or Negative.

The **buildFeatures** function of the **Features** class performs this task.

## 

## **B.2. Reviews to lines of tokens (Reviews -> Tokens)**

Here, each review is loaded to memory and cleaned for tokens. The obtained tokens are further cleaned by retaining only those tokens that are also in features we defined previously.

## 

## **B.3. Reviews to Bag-Of-Words Vectors (Reviews -> Tokens -> Vectors)**

## The data that is fed to the training model should be encoded into numerical values and all the training examples should be of uniform length. So far what we have is the training and test data in the textual form and of non-uniform length. We use the Bag-Of-Words model to encode the data in order to make it suitable for training/learning.

 In this model each review is transformed to an encoded vector where each word is assigned a score. The length of the vector corresponds to the length of the vocabulary. There are different methods for scoring the words. In this project we use '**count**' for scoring the words. Let's understand how it works.

**Example:**

Let’s assume that

Features = {this, that, is, mine, not, cat, dog}

text          = “This this is mine".

Then encoded text using the 'count' scoring method is

**[2, 0, 1, 1, 0, 0, 0]**

**Explanation:**

* The number of words in the text is 4.
* The length of the vector = length of the features = 7
* score of 'this' = (total occurrence of 'this' in the text) = 2
* the score of 'this' is stored in the index corresponding to 'this' in the vector(for now, we can assume it to be in the same index as in vocabulary).
* the words 'that', 'not', 'cat', 'dog' does not occur in the text so their corresponding indexes in the vector are assigned a score of zero

The process of **getWordVectorCount** of the **NaiveBayes** performs the complete task of review -> token -> count vector.

This count vector is fed into the model to train/predict the class to which the review belongs.

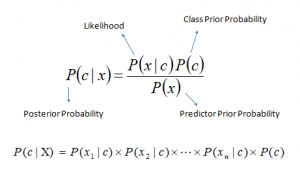
**C. SENTIMENT ANALYSIS MODEL**

**C.1. Building a Classifier Model**

There are different types of models that we can use. Here we have used **Naive Bayes Classifier.**

It is a classification technique based on **Bayes’ Theorem** with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other features.

 Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



Above,

* *P*(*c|x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
* *P*(*c*) is the prior probability of *class*.
* *P*(*x|c*) is the likelihood which is the probability of a predictor given *class*.
* *P*(*x*) is the prior probability of the predictor.

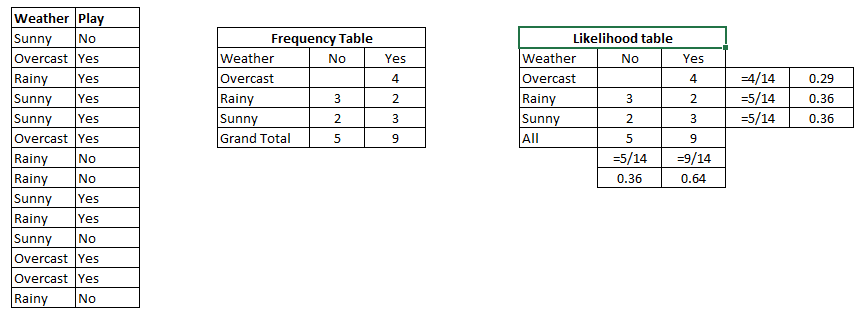
**How does Naive Bayes algorithm work?**

Let’s understand it using an example. Below I have a training data set of weather and corresponding target variable ‘Play’ (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather conditions. Let’s follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create a Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Step 3: Now, use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.



**Problem:**

Players will play if the weather is sunny. Is this statement correct?

We can solve it using the above discussed method of posterior probability.

P(Yes | Sunny) = P( Sunny | Yes) \* P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P( Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 \* 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different classes based on various attributes. This algorithm is mostly used in **text classification** and with problems having **multiple classes**

1. Outcome expected from the project:

Using the above Multinomial Naive Bayes model on the Stanford dataset we have achieved a maximum:

1. accuracy of 0.84
2. recall of 0.83
3. precision of 0.84
4. f-measure of 0.83

We achieved above figures on taking cut-off frequency = 70 (experimentally)

1. Summary of roles/responsibilities of all students:

|  |  |  |
| --- | --- | --- |
| Name | Registration Number | Contribution |
| Kritank Singh | 20174175 | Group Leader |
| Lakshay Saini | 20174106 | Member-1 |
| Avinash Sitaram | 20174168 | Member-2 |

Comments (if any):

Suggestions for improvement (if any): \_\_\_\_\_\_\_\_

Signature of Mentor

PANEL COMMENTS

Comments (if any):

Suggestions for improvement (if any): \_\_\_\_\_\_\_\_\_\_\_

Signature of Panel Representative