**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT iii**

**1. INTRODUCTIONARY CHAPTER 2**

1.1 INTRODUCTION 2

1.2 OBJECTIVE 3

1.3 LITERATURE SURVEY 3

**2. OUR METHODOLOGY 4**

2.1 FLOW CHART 4

2.2 FETCHING DATASET 4

2.3 CLEANING DATA 5

2.4 HAND TAGGING 5

2.5 TRAINING CLASSIFIER 6

2.6 TESTING CLASSIFIER 6

2.7 NAÏVE BAYES CLASSIFIER 7

3. **CONCLUSION** **8**

4. **REFERENCES 9**

**CHAPTER 1**

**INTRODUCTIONARY CHAPTER**

**1.1 INTRODUCTION**

Twitter is a microblogging media in real time to express the persuasion of a person or group about a particular topic to appear going on a timeline. A person can express his view in front of the world in various formslike multimedia, text etc. Thus huge and varied amount of knowledge can be extracted from the tweets.

In text mining, Sentiment Analysis and Opinion Mining consists study of sentiments, attitudes, reactions, evaluation of the content of the text.

Our goal in this project is to determine whether the expressed text (i.e., tweet) is happy, sad, angry, and neutral. It is also known as opinion mining, deriving the opinion or attitude of a speaker

Sentiment analysis is a challenge of the Natural Language Processing (NLP), text analytics and computational linguistics. In a general sense, sentiment analysis determines the opinion regarding the object/subject in discussion.

**1.2 OBJECTIVE**

The main objective of this project is to gain knowledge and experience in developing a real time application and also make best use of self-learnt subjects. Apart from this, to gain knowledge of Python, Machine Learning, Natural Language Processing and the way in which they can be used for our respective project.

Python is a popular Object Oriented Programming language and rich in its libraries which are useful in Machine Learning.

Machine Learning is the science of getting computers to act without being explicitly programmed.

Natural Language Processing Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken.

**1.3 LITERATURE SURVEY**

1. Geetika Gautam, Divakar Yadav “Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis”

This paper describes approaches specific to the Machine Learning.

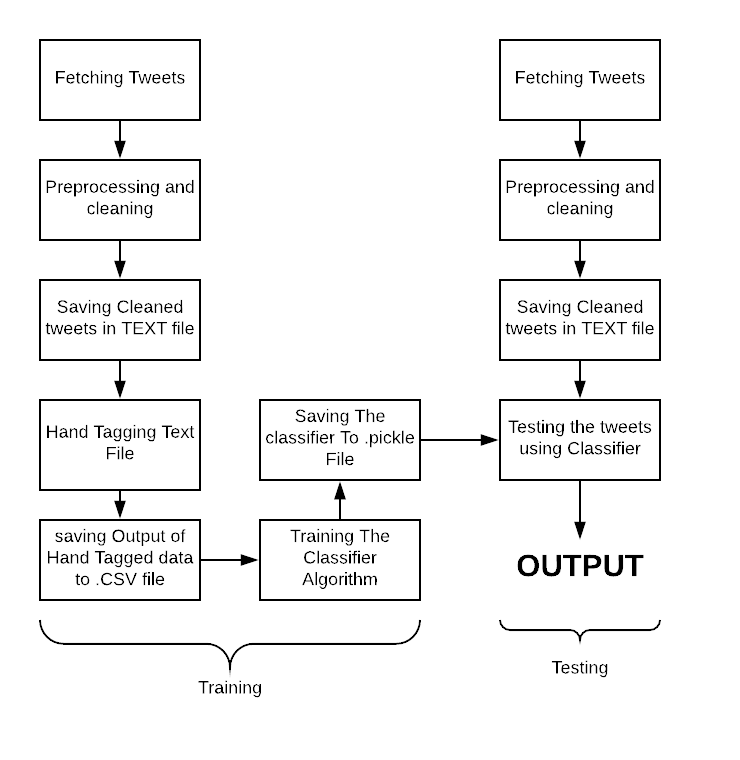
2. Harsh Thakkar, Dhiren Patel “Approaches for Sentiment Analysis on Twitter: A State-of-Art study”

This paper describes of approaches and methodologies for sentiment analysis.

**CHAPTER 2**

**OUR METHODOLOGY**

**2.1 FLOW CHART**



**2.2 FETCHING TWEETS**

The following steps are involved to fetch the tweets using twitter API:

1. Create own app on twitter using app.twitter.com

2. You will get 4 different keys that will help in authentication purpose.

3. Import the libraries from tweepy package.

4. To connect with twitter api we will use the function oauthhandler() and set\_access\_token()

5. Uses twitterstream to fetch the data (tweets).

6. Use twitterstream filter to filter the data (tweets) based on keyword and language.

**2.3 CLEANING DATA**

The following steps are involved to clean the tweets:

1. After fetching the tweets next comes pre-processing and cleaning the tweet.
2. Pre-processing cover the conversion from json format to text format and getting English tweets.
3. Cleaning of tweets covers plenty of things :
   1. Removing username and retweet
   2. Removing hyperlink
   3. Removing special character, numbers, whitespace, hashtags and emoji’s.
4. Saving file in text format for providing input to further steps.

**2.4 HAND TAGGING DATA**

The following steps are involved in hand tagging the tweets using GUI tool:

1. Input taken here is the “text file” of cleaned and pre-processed “tweets”.
2. CSV package is imported to use file-writer and reader.
3. Developed GUI tool using the package “tkinter” to simplify the task of hand tagging the dataset.
4. We are creating list of tuples and hand tagging it with multiple sentiments.
5. The hand tagged data is written in csv file.
6. This file, later will be used for further process of classification.

**2.5 TRAINING CLASSIFIER**

1. The hand tagged data from file is fed into the classifier to train to the algorithm.
2. We've used Naive Bayes Classifier which is based on Bayes Theorem.
3. Naïve Bayes classifier is used as it is fast, reliable, and accurate and works particularly well with natural language processing (NLP) problems.
4. Classifier is dumped to a .pickle extension file for future use.

**2.6 TESTING CLASSIFIER**

1. The cleaned and pre-processed tweets are read from the file.
2. Each and every tweet is then classified into its respective sentiment by the algorithm.
3. Then Percentage is calculated as per the sentiments classified by the algorithm as output.

**2.7 NAÏVE BAYES CLASSIFIER**

The simplest solutions are usually the most powerful ones, and Naïve Bayes is a good proof of that. In spite of the great advances of the Machine Learning in the last years, it has proven to not only be simple but also fast, accurate and reliable. It has been successfully used for many purposes, but it works particularly well with natural language processing (NLP) problems.

Naive Bayes is a family of probabilistic algorithms that take advantage of probability theory and Bayes’ Theorem to predict the category of a sample (like a piece of news or a customer review). They are probabilistic, which means that they calculate the probability of each category for a given sample, and then output the category with the highest one. The way they get these probabilities is by using Bayes’ Theorem, which describes the probability of a feature, based on prior knowledge of conditions that might be related to that feature.

**Bayes’ Theorem**

Now we need to transform the probability we want to calculate into something that can be calculated using word frequencies. For this, we will use some basic properties of probabilities, and Bayes’ Theorem.

Bayes’ Theorem is useful when working with conditional probabilities because it provides us with a way to reverse them:



**CHAPTER 3**

**CONCLUSION**

Open social networks are best examples of sociological trust. The exchange of messages, followers and friends and varying sentiments of users provide a crude platform to study behavioral trust in sentiment analysis domain. Machine learning approaches have been so far good in delivering accurate results. Depending upon the application, the success of any approach will vary. We observed a number of surprising and sometimes disheartening results. We were not able to achieve a high accuracy of predictions, most likely due to the nature of the dataset and the relative difficulty of selecting appropriate features to learn. On the other hand, we were able to learn a classifier that increased it is recall as the sample size grew. The key aim is to analyze a large amount of reviews by using twitter dataset which are already labeled by hand tagging. Naive Bayes classifiers are powerful tools for learning data, but as we have discovered they need to be excessively coddled and fine-tuned, if they are to be useful in all cases of performance.

**References:**

1. Learning Python Book by David Ascher and Mark Lutz
2. Approaches for Sentiment Analysis on Twitter: A State-of-Art study

by Harsh Thakkar and Dhiren Patel

1. Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis by Geetika Gautam Divakar Yadav
2. A practical explanation of a Naive Bayes classifier by Bruno Stecanella (<https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/>)