A MAJOR PROJECT-II REPORT
ON
Environmental Sentiment Analysis: Leveraging AI to Assess Public Perception of Ecological Issues Through Text Data Fusion

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY (M. TECH)

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

NIKHIL GURJAR

2K22/CSE/14

Under the Supervision of

Mr. Nipun Bansal

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042
MAY, 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **Nikhil Gurjar, 2K22/CSE/14** students of M.Tech, hereby declare that the project Dissertation titled "**Environmental Sentiment Analysis: Leveraging AI to Assess Public Perception of Ecological Issues Through Text Data Fusion**" which is submitted by me to the **Department of Computer Science and Engineering, Delhi Technological University, New Delhi** in partial fulfillment of the requirement for the award of degree of **Master of Technology**, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: **New Delhi** **Nikhil Gurjar**
Date: **2K22/CSE/14**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Environmental Sentiment Analysis: Leveraging AI to Assess Public Perception of Ecological Issues Through Text Data Fusion**" which is submitted by **Nikhil Gurjar, 2K22/CSE/14**, **Department of Computer Science and Engineering**, **Delhi Technological University, New Delhi** in partial fulfillment of the requirement for the award of the degree of **Master of Technology**, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: **New Delhi**
Date:

Mr. Nipun Bansal
Assistant Professor

ABSTRACT

Global warming, often referred to as climate change, is now emerging as one of among the most highly debated topics of over a decade or so. A lot of individuals think that warming temperatures pose a serious threat to our planet, even if some people claim it is a myth. This article examines how public opinions have changed over the last 10 years by using sentiment analysis to examine Twitter data. With 320 million active users each month, Twitter is a useful tool for determining public opinion. Using sentiment analysis, we extracted tweets that had terms like "global warming" and "climate change," classifying them according to whether they were neutral, positive, or negative. We trained numerous data sets utilizing Naïve Bayes, Multinomial Naïve Bayes equations and SVM-based classification algorithms for the purpose to reach highest possible accuracy. Then, employing data from Twitter, the approach with the highest accuracy rate has been employed to evaluate how perceptions on global warming have fluctuated over time.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) Bawana
Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to **Mr. Nipun Bansal** for his continuous guidance and mentorship that he provided us during the project. His direction helped me understand and accomplish my objectives, emphasizing the significance and industrial relevance of this project. He was always available to assist and clarify any doubts we encountered. This project would not have been successful without his continuous support and encouragement.

I would like to extend my sincere thanks to **"Dr. Vinod Kumar"**, *Head of Department*, Department of Computer Engineering, Delhi Technological University, New Delhi for his valuable suggestions and feedback on our project work.

Finally, I would like to thank all the participants who participated in the study, without whom this research would not have been possible. I express my sincere gratitude to all the individuals who have directly or indirectly contributed to the success of my project.

Nikhil Gurjar
(2K22/CSE/14)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The phenomenon of global warming, frequently referred to as the "greenhouse effect" or "climate change," is caused mostly by emissions from automobiles, manufacturing processes, and numerous other elements which soak up atmospheric greenhouse gasses that include carbon dioxide in the atmosphere of the planet. While released, these chemicals may explode beyond space, yet when their concentration grows, they become immobilized in the atmosphere of Earth. The aforementioned gasses, particularly the one called carbon dioxide (CO2), causes the surface temperature of the planet to go up because they absorb solar radiation. This process has resulted in a discernible rise in global temperatures during the last few decades, which is generally known as global-warming.

The melting of polar glaciers and an increase in Earth's average temperature brought on by global warming will cause sea levels to rise. Even while the great majority of experts concur that global warming is an urgent and genuine problem, some continue to question its veracity. Over time, the public's perception of global warming has changed as a result of political debate, scientific discoveries, and individual experiences.

Extreme weather phenomena like protracted heat waves, severe droughts, floods, and frequent storms have increased in frequency in the last several years. Warmer winters are causing frozen lakes to thaw sooner than expected. The years 2000–2009 were the warmest in the previous 1,300 years, according to researchers. The pattern indicates severe and potentially irreversible changes to Earth's climatic and geographical systems, with dry parts growing drier and wet ones becoming wetter. According to scientific predictions, Earth's temperature may increase by as much as 8 degrees Celsius by the year 2100.

With 328 million monthly active users, TAn internet-based social network i.e. Twitter has emerged as a useful tool for gauging public sentiment on a range of issues, including politics, entertainment, social issues as well as global warming. Ever since its launch, Twitter has offered a singular, instantaneous insight into the opinions of its users as a whole.

This study investigates common views on climate change and global warming using sentimental analysis on data collected from Twitter. The development and evaluation of several methods of classification on both general and climate change-specific Twitter datasets is part of the study. Identifying a particularly accurate approach to classify sentiments is the aim.

I've been recording five thousand tweets every year for the past ten years, and I utilize the most effective classifier to evaluate them and divide them into three separate sentiment classes: neutral, negative, and positive. This research aims to ascertain how, throughout the last 10 years , public perceptions of global warming have changed. For clarity, both the positive and neutral class definitions can be found below.

By the analysis of this data, this study seeks to enhance our understanding of public perceptions of global warming and how these perceptions have changed in reaction to new scientific findings and actual climate occurrences.

- Positive means : Individuals who concede and accept the existence of global warming.
- Negative means : Individuals that contest or deny the reality of global warming.
- Neutral means : Individuals who are neither in favor of nor against global warming are considered neutral.

The document's latter sections, mainly follow the opening statement, dive into more information into the numerous sentiment detection classification frameworks and evaluation measurements. The Multinomial Naive Bayes classification algorithm and Naïve Bayes from the

NLTK platforms are reviewed in the second section after several SVM versions available in the Scikit platform are explored. Moreover, the tool of methodologies such as stop words, N-Gram Analysis, Tweepy-Analysis and the occurrence frequency of terms and the Inverse-Document-Frequency (TF-IDF) are explored for enhancing sentiment analysis precision.

The processes needed for gathering and assessing information are outlined in the third component, providing researchers an exhaustive manual. The methodology for testing is explained thoroughly in the final section, in addition to the approaches that are used to assess how well different classifiers perform.

In the fifth part, test data and graphs are used to illustrate how well the NLTK Naive Bayes classifier performs in sentiment classification. The outcomes of the SVM and Scikit Multinomial Naïve Bayes classifier tests are then thoroughly analyzed in the sixth and seventh sections, respectively.

The conclusions obtained from each trial are merged in the eighth some way, which in addition compares the success rates with various classifiers and chooses the best classifications strategy. The following subsection provides an extensive examination of several assessments of sentiment strategies.

The results of applying the classifier with the greatest accuracy rate to categorize climate change-related Twitter data during the previous ten years are shown in the ninth section. This section provides insightful information about how public attitudes toward climate change have changed throughout time.

The study's conclusions are presented in the tenth part, which also suggests future directions for research on sentiment analysis and perceptions of climate change.

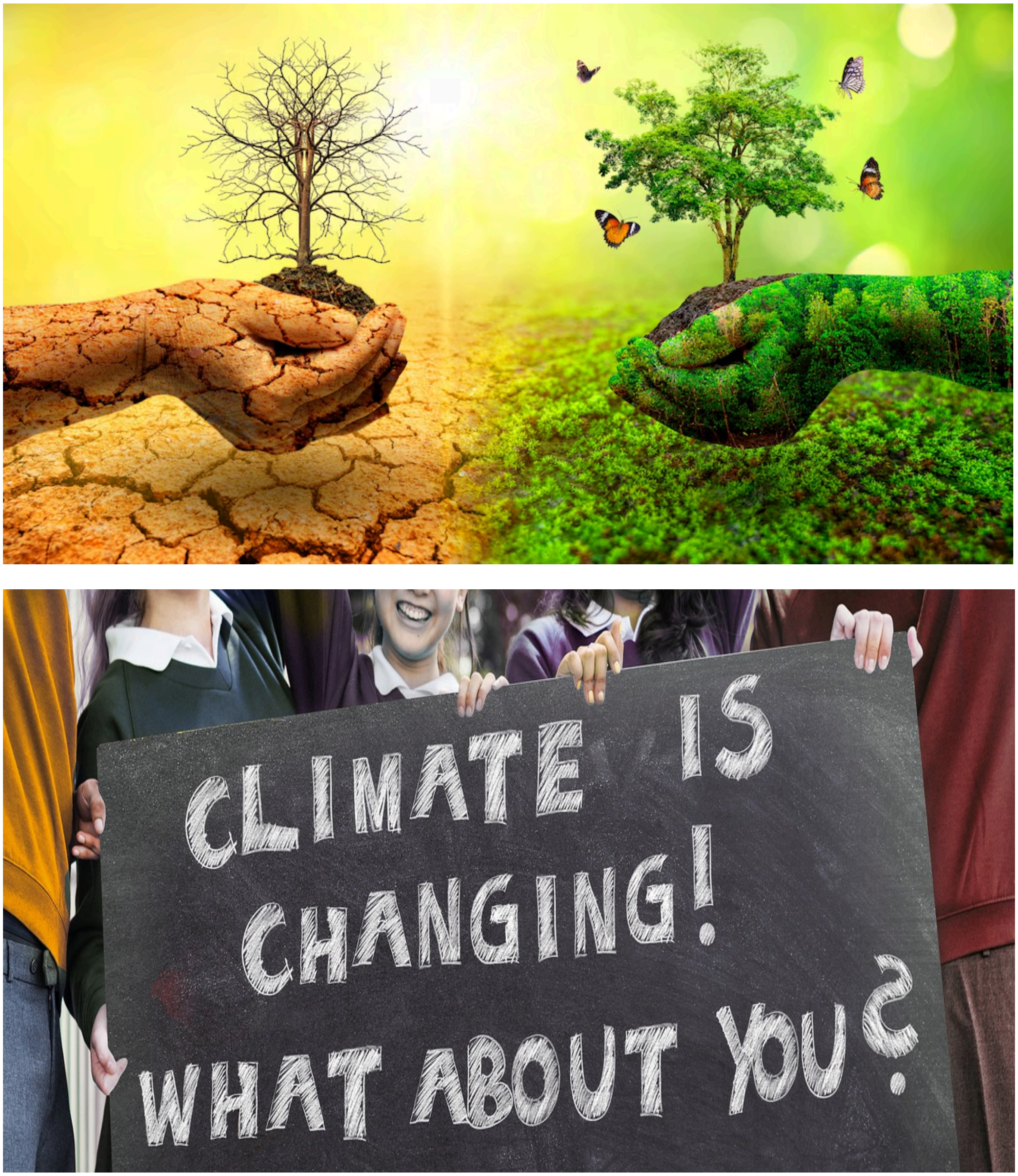


Fig 0 : Climate Change

2. LITERATURE REVIEW

An exhaustive summary of the tools and techniques employed for the present study is given in this section. It addresses the different characteristics and procedures used in current, and applicable studies.

The relevance of recognizing the technologies and methodologies used in sentiment analysis is pointed out in the paper. It underlines how crucial it is to remain kept up with current research findings and techniques With the objective to guarantee the suitability and effectiveness of the research.

This study covers relevant literature and looks at a variety of factors along with methods that are frequently utilized in studies addressing sentiment analysis. This covers an examination of the benefits and downsides of multiple approaches besides delivering insights into fresh developments and patterns in the industry.

The literature review gives guidance regarding a particular study's methodological choices by integrating information obtained from past research. It boosts the reliability and credibility of the researched findings by offering a strong basis for comprehending the reasoning behind the chosen technology and methodology.

All things considered, the literature review serves a vital function in the research study after providing insightful assessment and direction for the creation and execution of the present sentiment analysis study.

2.1. Tweepy

The Python module named Tweepy, which is brought up in [4], makes it quicker for Python programs to share information with Twitter's programming interface for applications (API). The technique for acquiring historical and current Twitter data is streamlined by this integration. With Tweepy, users can use Twitter for a variety of tasks by using its features when combined with various other libraries designed by developers worldwide [6].

2.2. Naïve Bayes Classifier

One fundamental method of classification which is frequently utilized in AI is the Simple Bayes classifier.It is a constituent of the natural language processing toolkit (NLTK) and functions particularly effectively for linguistic analysis of content [5].

To allow for this classifier to operate effectively, each document goes through, and the possibility that each keyword will be assigned to a good, negative, or neutral behavior is assessed. Afterwards, it analyzes the probability with the labeled sentiment in the tweet. The classification algorithm has its foundation on the basic concept of the Bayes theorem, which states that every variable is treated as independent. This implies that every aspect of the text is assessed independently of the others.

Large datasets are easy to handle for the Naïve Bayes classifier, therefore rendering it an excellent pick for analyzing feelings use cases. It performs well in classification with multiple classes scenarios and functions well in real-time environments.For sentiment analysis, the naive Bayes classification algorithm is a useful tool on tweets on the environment because of its efficiency in assessing sentiments on social media sites like Twitter.

The mathematical foundation for the functioning of this classifier is the Bayes theorem, as expounded in references [17][20].

Here is a definition of the concepts mentioned above:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

The conditional probability of event A happening given the occurrence of event B is denoted as P(A|B). On the other hand, P(B|A) represents the likelihood of event B happening when event A has already taken place. P(A) reflects the probability of event A happening, whereas P(B) signifies the chance of event B occurring.

2.3. Multinomial Naïve-Bayes Classification

One sophisticated tool in the SciKit-learn package is the Multinomial Naïve-Bayesian classification technique. This technique outperforms the conventional Naive Bayes method in terms of effectiveness and produces better results. The traditional Naive Bayes approach assesses each characteristic independently, while multinomial categorizers regard each category as independent from the others.

Based on the likelihood that one of those features might be found in a particular category, this probability method of classification estimates the probability that a class will show up throughout the content. The way the Multinomial Naïve-Bayesian classifier and the traditional Naïve Bayes classifier handle class independence is the primary disparity between them.

The classifier Multinomial Naïve Bayes was used in present study, and made use of numerous iterations of n- Grams, removal of stopping words, and an alpha smoothing factor. Section 6 of the study presents the classifier's results.

Further details about the use of the Multinomial Naive Bayes classification algorithm are provided in Reference [16].

$$\mathbf{P(C/D)} \quad \boldsymbol{\alpha} \quad \mathbf{P(C)} \prod_{k=1}^{k<=n_k} P(\frac{t_c}{k})$$

P(C|D) is the probability of class C given the documentation D. P(C) is the prior probability that a text or tweet is a member of class C. The variable "frequency" denotes how often a certain term occurs. The conditional probability of the word "t" is denoted by P(tk).

2.4. Support Vector Machines

Effective machine learning methods which perform effectively in instances of use combining classification and regression consist of support vector machines. SVM has become commonplace in employment opportunities needing categorisation. By establishing the most effective hyperplane between multiple categories, it breaks them down. Numerous support vector machine kernel functions can be employed to construct an appropriate model for determining the hyperplane. Poly, Linear, Sigmoid, and RBF comprise the core functions. Using every single kernel function, I have identified the optimal definition hyperplane for this study. This work also makes use of classifications of C - support vectors and linear support-vectors.. Support vector classification using linear methodology has similarities to support vector categorization utilizing a linear technique. But linear machine learning with support vectors provides better flexibility in terms of adjusting for fines and losses. As a result, as noted in [18], larger data sets provide better classification performance with linear support vectors.[20][28].

2.5. Parameters

Stopping words, n- Grams, and TF-IDF are parameters of interest that are implemented when combined with The support vector machine classifiers include the Naive Bayes method, the Multinomial Naïve-Bayes, and others. Specifically, experiments' outcomes are significantly influenced by additive smoothing when these factors are used.

2.5.1. N-gram

Creating the distribution of probabilities for a particular word pair is an aspect that constitutes language modeling. One approach for the modeling of languages that separates documents of text into a series of associated words is the N- Gram. The information contained in the content of the document is separated into two groups: N- Grams and The N- Gram pairings. Report Word The algorithms will be trained using these n-gram collections.

2.5.1.1. Unigram

Words may be employed to separate written material into segments. In an n- Gram, the letter "n" designates the dimension of the gram. A gram, which is the size of one, is commonly referred to as a unigram. Consider the next sentence, for instance: "The ocean's heat is rising as a consequence of global weather change."

The text can be broken down into the parts that follow using the unigram technique: "The impact of climate change is leading to a rise in ocean temperature." Phrases like "is," "in," "are," and other comparable terms could contribute barely any impact to the sentiment analysis of probability distributions. Bigram and trigram possibilities are better solutions for a wider variety of feature vectors.

Ex: the surface temperature of the ocean is going up due to worldwide warming.

The preceding statement can be split into the parts that follow using a unigram: "The impact of climate change is leading to a rise in ocean levels.""

Sentiment analysis's probability distributions gain little from words like "is," "in," "are," and related ones. Bigram and trigram are excellent choices if you need a wider range of feature vectors.

2.5.1.2. Bigram

When n equals two, the term "bigram" is used. Consider the subsequent sentence, for instance : "Climate change is causing the ocean's temperature to rise."

The text may be divided into the following pairs of successive words using a bigram approach:Ocean temperatures are determined by the following: (ocean, increase), (rise, in), (in, ocean), (change, climate), (is, causing), and so on. A more thorough examination of the text's context and organization is made possible by this segmentation.

2.5.1.3. Trigram

When n equals three, the result is a trigram. Take the following sentence, for example: "The ocean's temperature is rising due to climate change."

The following word order can be seen in a trigram analysis:Ocean temperatures are determined by the following: (ocean, increase), (rise, in), (in, ocean), (change, climate), (is, causing), and so on. This dissection offers a more profound comprehension of the context and organization of the text.

2.5.1.4. Four-gram

An n that is equivalent to four is called a four-gram. For instance, take the sentence: "The temperature increased in the ocean and sea, the climate changed and is producing a rise in the temperature, and the change is creating a rise in the temperature."

The text can be divided into groups of four consecutive words in a four-gram analysis. This division makes it possible to analyze the structure and background of the text in great depth.

2.5.1.5. Five-gram

The size is referred to as a five-gram when n = 5. Take the following sentence, for example: "The ocean's temperature is rising due to climate change."

The sentence can be broken down into a technique for determining the relative importance of phrases inside a given document is called TF-IDF, or Term Frequency-Inverse Document Frequency the following five-word sequences using a five-gram analysis:

1. (is, contributing, to, the, ocean's)
2. (temperature, rise, altering, is, producing, rising) in the ocean

3. (Rise, is, caused, by, climatic change)

A thorough analysis of the text's context and organization is made possible by this segmentation. A method for determining the importance of words in a given text is called Term Frequency-Inverse Document Frequency (TF-IDF) which is used to determine the relevance of terms inside a text by assessing their frequency and the inverse frequency across a group of documents. For ascertaining the relevance of each phrase, a simple categorization method is used. Stop words that don't significantly add to sentiment analysis, including "is," "it," "that," and "them," are eliminated throughout this phase in order to precisely define each term's weight.

The Multinomial Naïve-Bayes classifier was used in this study in conjunction with TF-IDF. Even though tweets are more often single words than long documents, the precision is not significantly impacted by adjustments in the Minimum frequency of use or frequency of documents. No appreciable increases in accuracy were seen despite many tries with Using document frequencies and minimum term frequencies adjusted between 5 and 2000.

TF-IDF produces strong results in long texts with a lot of word repetition. In these situations, it is very useful. The computation of TF(term) involves the no. of phrases in the paper by the quantity of times the word has occurred, whereas IDF(term) is determined by having an inverse document frequency (IDF) may be calculated by dividing the total occurrences a phrase appears in the documents by the logarithmic-mean of the entire amount of documents.

2.5.2. Stopwords

By lowering computing cost, removing stop words from text can greatly improve the classification process. Stop words sometimes cause needless complication in categorization assignments since they usually have little or no relevance in a phrase. Eliminating these terms from tweets can increase precision overall.

2.5.3. Additive-Smoothing

Laplace & Lidstone smoothing are used in additive-smoothing, which is used to halt classification overfitting. When a word comes up during testing or validation, the algorithm's effectiveness in categorizing it during training could not translate. This problem and the missing data are taken into consideration by adding the smoothing value alpha to the calculation. The size and completeness of the dataset determine how much alpha is needed to cover all possible attributes.

$$P_{lap} = \frac{X_{i+\alpha}}{N + \alpha_{d}}$$

"Lidstone smoothing" describes the procedure when α is smaller than 1. On the other hand, Laplace smoothing is used when α is either equal to or larger than 1.

In computational linguistics and natural language processing (NLP), N-grams—contiguous sequences of n elements—such as words, characters, or symbols—are retrieved from a given text or audio sample and are essential components. These sequences aid in the comprehension and analysis of the patterns and structure seen in textual material. For example, N-grams can be used in language modeling to anticipate the following word in a phrase by taking into account the (n-1) words that came before it. For applications like text production, where coherent and contextually relevant text generation is required, this predictive power is essential. Moreover, N-grams play a crucial role in information retrieval systems, facilitating the effective indexing and searching of extensive text collections. N-grams are very useful in the R programming language for applications like sentiment analysis and text mining, where finding recurring patterns or trends in text can yield insightful information. Researchers and developers may create models that accurately represent the subtleties of human language by utilizing N-grams, which will ultimately improve the efficiency and precision of NLP applications.

N-grams are used in text creation to generate new text by taking a sample from the probability distribution of N-grams found in a corpus. Applications such as chatbots, poetry creation, and automated essay writing can make use of this technology. Models may be trained to produce text that closely resembles the original data's style and organization by examining the frequency of N-grams in a sizable dataset.

3. DATA COLLECTION PROCESS AND HANDLING PROCESS

3.1. Data Collection Process

By making connections between hardware, software, and web applications easy, the Twitter streaming API improves data collection. We need the Application Programming Interface (API) key, Application Programming Interface (API) secret, access-token, and access-token-secret in order to access this Application Programming Interface (API).

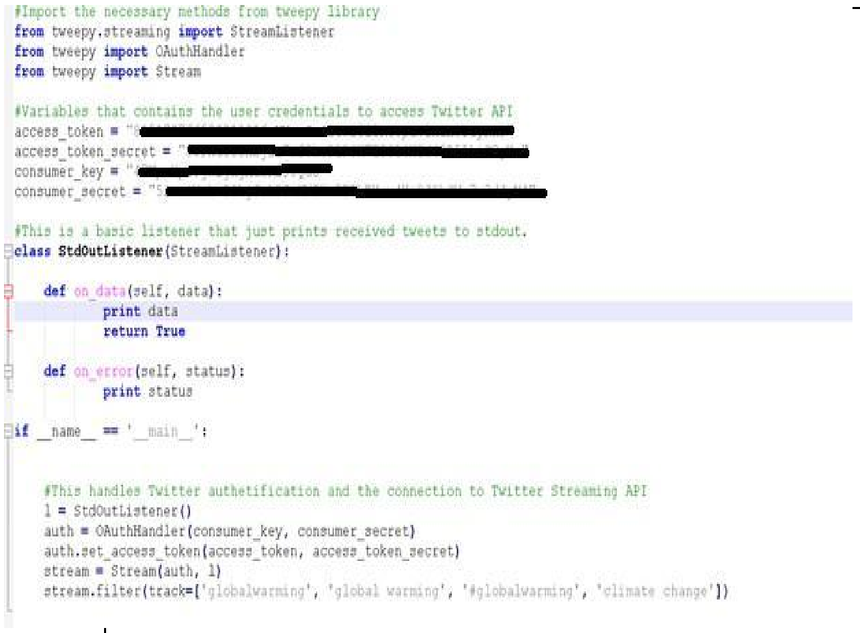
The following procedures will let you get the keys needed to use the Twitter streaming API:

Firstly, if you haven't already, register on Twitter.

2. Visit apps.twitter.com and enter your Twitter login credentials.
3. Take into account creating a fresh application.
4. Complete the "Create a New Twitter Application" form.
5. Acquire the secret and API key.
6. Obtain the access token and the secret that goes with it.

After obtaining all four keys, you may retrieve tweets using the Tweepy Python application. Because Tweepy is integrated with the Twitter streaming API, you may get tweets that are pertinent to climate change. Tweets that specifically mentioned For this experiment, For this experiment, data on " global warming " and the "impact of climate change" were gathered.

An example of code that demonstrates how to get tweets from Twitter on global warming may be seen below [5] [23].



```
#Import the necessary methods from tweepy library
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream

#Variables that contains the user credentials to access Twitter API
access_token = "[redacted]
access_token_secret = "[redacted]
consumer_key = "[redacted]
consumer_secret = "[redacted]

#This is a basic listener that just prints received tweets to stdout.
class StdOutListener(StreamListener):

    def on_data(self, data):
            print data
            return True

    def on_error(self, status):
            print status

if __name__ == '__main__':


    #This handles Twitter authetification and the connection to Twitter Streaming API
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, l)
    stream.filter(track=['globalwarming', 'global warming', '#globalwarming', 'climate change'])
```

Figure 1. Code Tweet Retrieval Example

3.2. Data-Processing

There are three types (positive, negative, and neutral) for each tweet that is kept in a CSV file. Table 1 provides examples showing how tweets are classified into different groups.

Table 1. Example of Classifying Tweets

Tweet	Sentiment
You stupid Democrats, There is no such thing as climate change.	Negative
Whether climate change could impact the quality of our water #It's Time to Change the Climate: https://t.com/z7AJT0apltFR Accompany us on @ZERO_CO2	Positive

According to the creator of Weather Channel, there is no such thing as climate change.	Negative
However, the climate change idea is untrue. Nothing is changing!	Negative

Trump may now pass laws that would offer universal health care, reduce global warming, and regulate immigration.	Positive
Californian Thinks 'Healthy Soil' Will Help Combat Global Warming	Positive
Regarding #climate change and \"Extreme Weather\," YOU have been lied to. For complete proof, view the data here. This link: t.co/kWaCLwMZ4e through @Ju2026	Neutral
To all Agmore residents who are interested in providing financing for projects related to the environment, climate change, and the countryside	Positive
Why current climate change is not caused by solar eclipses	Neutral
Climate change costs India $10 billion annually	Positive
As climate change accelerates, Berlins' infrastructure is adapting to address drainage and heat-related issues.	Positive
The melting of Arctic sea ice is the reason for the initial Pacific regions Walrus haul out in history: #climate_change #its_time_to_change https://t.com/RzvS67eQdvVV Take a look at @ZERO_CO2_	Positive
The people most impacted by climate change may be women. #Action_Against_Climate #Preserve It t.co/1Itnump6fSy is the URL.	Positive

3.2.1. Data Preprocessing

Each tweet passes through the subsequent preliminary processing steps:

1. All of the words should be lowercase.

For example: "Climate Change: A Healthier Society is completely Realistic Considering the Massive Challenge"

"Climate change for a better world is 100% feasible," the conclusion states.

2. Use "LINK" in lieu of links that finish in "http" or "https".

Example: "Letting #climatechange continue is inexplicable. The following details are accurate: [t.co/L0WW9xWT7K]"

Result: "Letting climate change continue is inexplicable. These are some details. Link"

3. Use "USER_REF" in lieu of "@Usernames".

For instance, "@LamarSmithTX21 It is inconceivable to allow #climatechange to continue. These are some details. t.co/L0RWW9xWT%7K is the URL.

Outcome: "USER_REFERENCE" Here are some facts on climate change that you may find incomprehensible: LINKS.

4. Remove all the white-spaces from the messages.

As an example, "@LamarrSmithTrX21 It is incomprehensible to allow #climate_change to continue. Here is a rundown of the details: [t.co/L0WW59xWT7K]

What's produced is "USER_REFERENCE." It is incomprehensible that We are permitting climate change to persist.

These are some of the specifics: LINKS

5. Remove every hashtag from the messages.

Example: "Letting #climatechange continue is inexplicable. This is a summary of facts: The URL is t.co/L0WW9xWT7K.

The outcome is "USER_REF." The fact that we are putting up with climate change is inexplicable. Here are a few details LINK.

6. Take off every comma from each tweet.

For example: "Climate_Change: A Healthier Society is entirely POSSIBLE, Considering the Huge obstacle."

Summary: "Climate_Change: A Healthier Society is entirely POSSIBLE, Considering the Huge obstacle.

Regular expressions and Python's string manipulation tools were used to create these modifications. Text was converted to lowercase using the lowercase() function, and URLs, usernames, hashtags, and white spaces were replaced with the proper labels using theuse the regular expression class's sub() function. Finally, the remaining punctuation in the text was eliminated via the strip() method.

The NLTK and Scikit libraries provide the stop words list easily accessible. In addition, "USER_REF" and "LINK" have been added to the list. One stop word per line in a text file may be used to create custom stop words. Each line is added to a stopwords list data structure once the file has been read; this data structure will be utilized in later stages of the process.

3.2.2. Stemming

In order to minimize index files, stemming entails finding derivative words and giving each derived word a common term. Using the NLTK Tokenizer Package in Python, the original tweet's The string's structure is converted into a list of substrings, allowing for the finding of every word and punctuation. To prevent issues with encoded strings, the original string is decoded to utf8. The NLTK library's nltk.stem.porter function already incorporates the Porter stemmer approach applied to the tokenized string in the module.

To split text into lists of substrings, tokenizers are used. Tokenizers, for instance, are able to separate words and punctuation from a string. As shown [32], the terms in the tweets are standardized using the Porter stemming approach.

3.2.3. Big Picture

Every tweet is stored in a text file within a written document ended in emotion and text. These files are iterated over using the Python data analysis function read_csv() from the pandas module, which records the file type and separator. These text files are used to extract features using the feature extraction module of SCIKIT.

Tweets(Posts) are converted through a token matrix counts using the vectorizer for counts approach (feature.extraction.text.CountVectorizer). The previously mentioned techniques are used to apply preprocessing stages including Stop word, tokenizing, and stemming, n- Gram building, and minimum_df choice. This method builds required preprocessors and analyzers'.

The method tfidf_transformer, which is part of the feature extraction module, uses the count-matrix to produce a scaled tf / tf-idf depiction that the count vectorizer generates. This tweet representation and the emotion that goes along with it are used by the classification algorithm to train the model.

4. TEST METHODOLOGY

This' study uses Twitter data in performing sentimental-analysis on global warming. Support-vector machines and Naïve-Bayes classifiers use stop-words, TF- IDF, N- Gram parameters, and multinomial Naïve Bayes classifiers. Each classifier is trained using two Twitter datasets, and the accuracy is evaluated using a test dataset.

Every year over the last ten years, 5,000 tweets have been gathered to train the algorithm and track changes in popular perception of global warming. Sentiment analysis and historical climate change data analysis are performed by the ideal classifier, which has been trained on multiple parameters and optimized.

4.1. Training Data

In this study, two datasets were employed.Twenty thousand tweets make up the first dataset, which was assembled from a number of publicly accessible datasets as cited in references [23, 24, 25]. 18,000 of these tweets are reserved for training. The second dataset comprises 10,000 tweets on climate change, of which 8,000 were taken from Twitter, reviewed by hand, and given sentiment annotations.

While tweets on global warming and climate change are particularly targeted in the second dataset, tweets with a broad emphasis are included in the first dataset. The training dataset has been labeled with sentiments that are positive-case, negative-case and neutral-case.

4.2. Testing Data

After the algorithms have been trained on the training datasets, experiments are carried out utilizing distinct test datasets. There are two thousand testing and validation datasets for every dataset listed in Section 4.1.

4.3. Randomness

There is a degree of randomness in the process of choosing the training and test datasets. Take the dataset unrelated to climate change,This involves 18,000 tweets in the training set and 2,000 tweets in the assessment sample. The Python-programming-package "random_sample (x_range(1, 21000), 19000)" , is used to acquire the training dataset. Out of the dataset, 18,000 tweets are chosen at random using this algorithm. The testing dataset then consists of the last 2,000 tweets.

The technique revisits over three independently generated separate data sets for training and testing to determine maximize performance that are taken from the 20,000 tweet. The same procedure is used to tweets that are generally focused and those that are explicitly dedicated to climate change.

5. RESULTS: NAÏVE BAYES CLASSIFICATION

The Naïve Bayes classifier was assessed using the sentiment analysis dataset from Twitter.

5.1. Unigram Implementation

I used the Unigram method and eliminated stop words while creating the training and testing datasets. This technique recorded a training duration of 870 seconds and produced an accuracy rate of 53.10%. The reliability and effectiveness of the analysis carried out were guaranteed by this painstaking procedure.

5.2. Bigram Implementation

Stop words have been eliminated and Bigram analysis using the datasets for testing and training.The training procedure took around 6686 seconds, yielding a 66.8% accuracy rate.

In order to compile more thorough contextual data, the computer looked at bigrams, or pairings of neighboring words, in the tweets during the training phase. In order to concentrate on keywords with more significance, stop words like "is," "the," and "and" were also taken out of the analysis.

The attained accuracy of 66.8% indicates the efficacy of the bigram analysis technique in capturing the sentiment subtleties within the Twitter data, despite the substantial computing effort needed for training. This procedure helps in identifying how attitudes and beliefs about climate change are evolving over time.

5.3. Test Accuracy

Using the Naïve Bayes classifier and stop word removal, the bigram analysis produced an accuracy of 66.9%. In contrast, 53.10% accuracy was obtained when the Naïve Bayes classifier was used just with unigrams.

Bigrams, which take into account pairs of neighboring words, have been shown to be more successful in gathering contextual information and enhancing sentiment analysis accuracy. This increased accuracy shows that word pair analysis may provide more accurate classification results by better capturing the complex emotion portrayed in the tweets.

Table 2. Bi-Gram NLTK vs Uni-Gram

NLTK(Combiation)	**Percentage-Accuracy**	**Time for Training(Seconds)**
Unigram+Stop words	53.10%	870
Bigram+stop words	65.8	6686

Table 2 shows that the implementation of a bigram is over eight times faster than the implementation of a unigram. Long run durations would be a barrier to getting the optimum time to results since machine learning is an iterative process.

Table 3: The Best-Case Summary Step-1

Percentage-of-Accuracy	65.8((Bigram + Naïve Bayes)
Training Duration (in seconds)	6686 Sec

6. RESULTS: MULTINOMIAL NAÏVE BAYES CLASSIFIER

Employing a dataset for sentiment analysis by having twenty thousand tweets from Twitter, I evaluated the Multinomial Naïve Bayes classifier. 18,000 randomly chosen tweets from the dataset made up the training set for each run, while the remaining 2,000 tweets made up the test dataset.

I explored various combinations of n-grams, including single words (uni-grams), pairs of words (bi-grams), and triplets of words (tri-grams), four words (four-Grams), and five words (five-Grams), in order to evaluate scikit-Multinomial Naïve Bayes algorithm, which was trained on 18,000 tweets. The obtained accuracy percentages varied between 21.24% and 55.5%. The unigram method produced the best accuracy of these, coming in at 55.5%. Bigrams and unigrams together had an accuracy of 50.25%.

In this specific experiment, unigrams performed better than other n-gram combinations, highlighting the significance of taking this into account while doing sentiment analysis tasks.

6.1. N-gram Iterations

Table 4: Index-Description

value of n (in n- gram)	Description
one	Uni-grams
two	Bi-grams
three	Tri-grams
four	Four-grams
five	Five-grams
one, two	Unigrams + Bigrams
one, two, three	Unigrams + Bigrams + Trigrams
one, two, three, four	Unigrams + Bigrams +Trigrams +Four-grams

Table 4. Index-Description(Continued)

n-gram	**Description**
One, two, three, four, five	Unigrams + Bigrams + Trigrams + Four-grams + Five-grams
two, three, four, five	Bigrams + Trigrams + Four-grams + Five-grams
three, four, five	Trigrams + Four-grams + Five-grams
four, five	Four-grams + Five-grams

The results of each combination chosen for the multinomial naive-Bayesian approach utilizing n-Gram are shown in Figure 2 below.

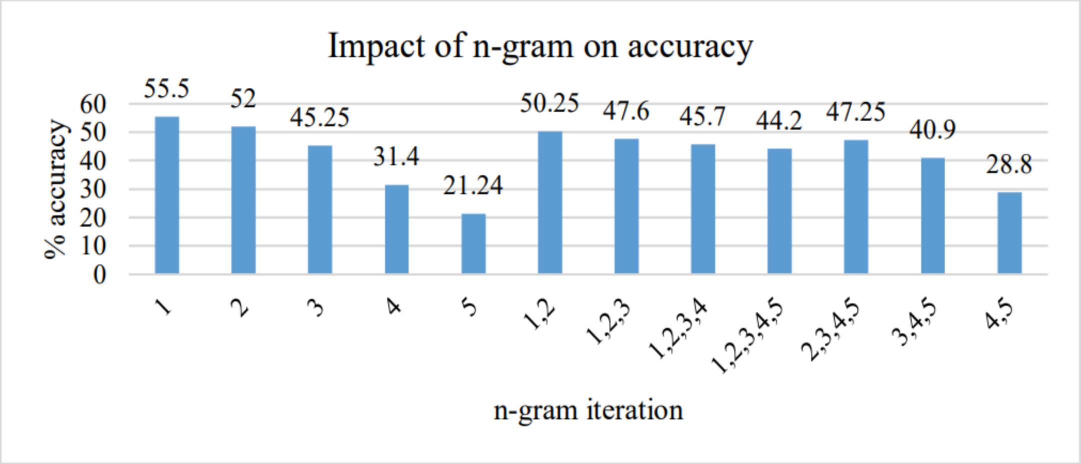


Figure 2: Effect of n-gram on Multinomial NB

Table 5: Best Case Summary Step 2 Accuracy

Percentage-of-Accuracy	55.50%
Algorithm-Employed	Multi-nomial Naïve Bayes+unigram

6.2. N-gram Iterations

An accuracy of 67.1% was attained after eliminating stop words from the training dataset and then using an α= 0.05 smoothing factor. Stop words came from a bespoke collection as well as the NLTK and Scikit libraries. Remarkably, comparable outcomes were attained irrespective of the origin of stop words.

Bigram and unigram together produced the best accuracy in this test scenario, coming in at 67.1%, over the prior accuracy rating of 55.5%. The accuracy results for all chosen combinations with the smoothing factor are shown in the graph in Figure 3.

Similar accuracy ratings were obtained when stop words from the NLTK or Scikit platforms were compared with bespoke stop words, as shown in Figure 3. This demonstrates how well the selected strategy worked to provide consistent outcomes across various stop word collections.

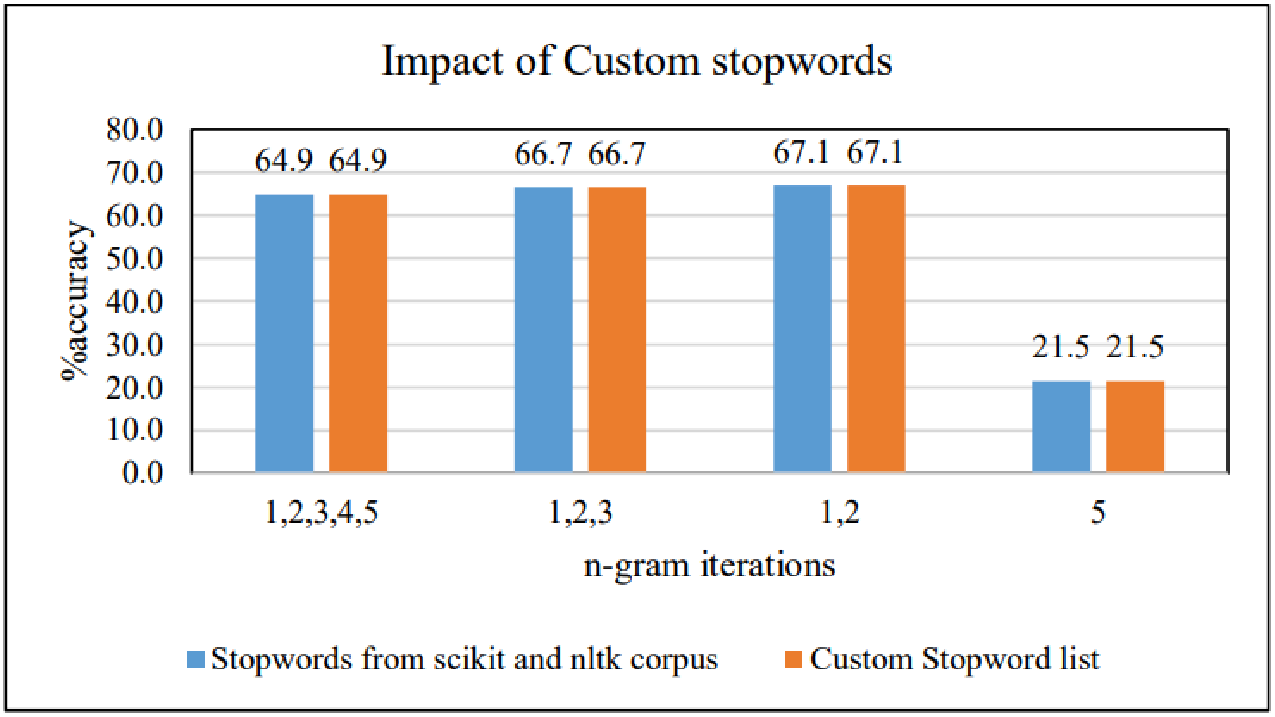


Figure 3: NLTK Corpus Stopwords vs. Custom Stopwords

Table 6: Step 3 of the Best Case Summary

Percentage-of-Accuracy	66.7
Algorithm-Employed	the MNB approach + Personalized halt/stop words, NLTK & Scikit MNB + halt/stop words

6.3. The N- Gram Iterations Using Alpha Smoothing Parameter with TF-IDF

After using the smoothing factor alpha in the Multinomial Naïve Bayes approach once again, I carried out the same procedure with n-gram combinations, including bi-phrases, tri-phrases, quartet-phrases, and quintet-phrases. The accuracy obtained by combining the bigram and unigram techniques was 67.05 percent.

The graph below shows the outcomes of several combinations I attempted using a Laplace smoothing factor set at an alpha (α) value of 0.04.

The graph shows every combination that has been tried and the accompanying outcomes.

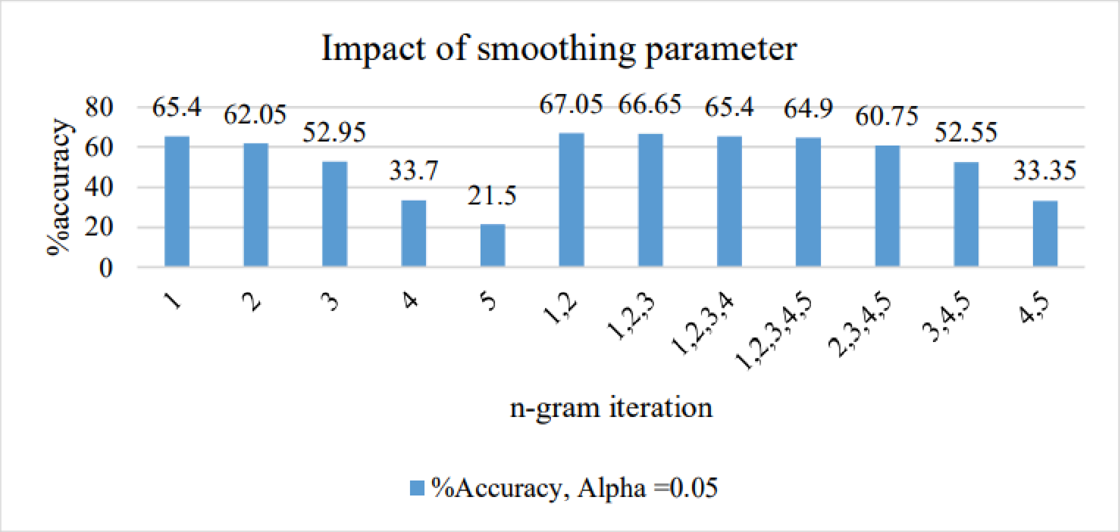


Figure 4: The Effect of Alpha, the Smoothing Parameter, on Accuracy

Table 7: Step 4 of the Best Case Summary

Percentage-of-Accuracy	65.14
Algorithm-Employed	MNB + halt_phrases + Bigram + alpha=0.05 + Unigram + TF-IDF

6.4. Tuning Alpha Parameter Value

After eliminating stop terms and raising the alpha coefficient 1 from 0.01, accuracy increased.
The updated accuracies were attained with an alpha value of 67.3. On the other hand, when the alpha value was lowered from 0.04 to 1, accuracy levels dropped.

The accuracy rates achieved for a variety of alpha values are shown in Figure 5 below, emphasizing the effect of changing alpha values on the classification model’s precision.

Table 8. Best Case Summary Step 5

Percentage-of-Accuracy	66.2
Algorithm-Employed	Unigram + Bigram along with MNB, α(0.04)

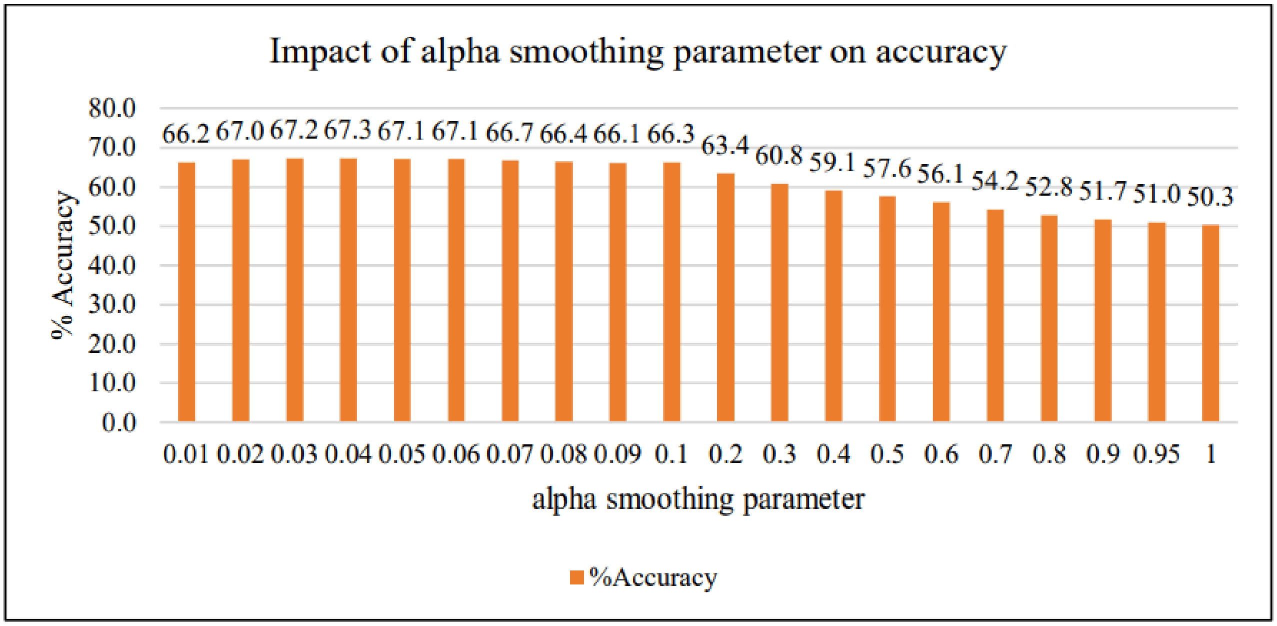


Figure 5: Tuning Smoothing Parameter

7. SUPPORT VECTOR MACHINES

Twenty thousand tweets from a Twitter sentiment analysis dataset were used for this study. The best three runs are shown below, with 18,000 randomly chosen tweets from the dataset as the training set and the remaining 2,000 tweets forming the test dataset.

Two different kinds of support vector machine techniques were used in this section:

1. Four different Kernel techniques for Support Vector Classification (SVC):
   - ❖ Polynomial
   - ❖ RBF
   - ❖ Sigmoid
   - ❖ Linear
2. Classification using Linear Support Vectors (Linear - SVC)

7.1. Support Vector Classification -Linear Classifier

Below are the outcomes of using the Linear kernel in numerous rounds of Support Vector Classification (SVC). The combination of SVC with Bigram and Unigram produced the greatest accuracy percentage (70.5%) out of all the iterations. This result is shown in Figure 6 by showing how well the SVC (Linear kernel) technique works in conjunction with the bigram and unigram feature analyses.

Table 9. Best Case Summary Step 6

Percentage-of-Accuracy	69.4
Algorithm-Employed	Support Vector Classification (SVC) using Bigram, Unigram, Linear Method, and stop words
Training Duration (in seconds)	586 sec
Testing Duration (in seconds)	5.972 sec

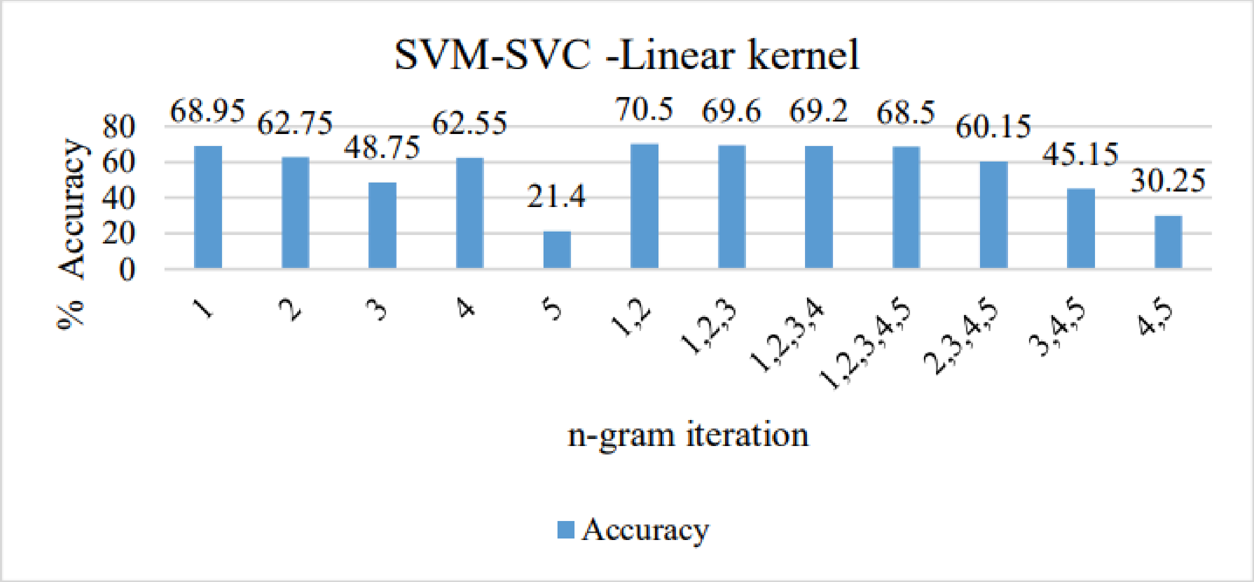


Figure 6: N-gram's Effect on SVM-SVC-Linear Kernel Accuracy

Table 9 shows that this specific case's runtime is more than 10 times faster than the NLTK Naïve Bayesian and bigram scenario displayed in table 4.

We looked at the Unigram and bigram combo using the sigmoid, RBF, and polynomial algorithms because of how well it worked with the linear method. The results are shown in Table 10 below, and it is consistently the case that the linear kernel outperforms all other evaluated kernel.

Table 10. Effect of Various Kernels in SVC on Accuracy

Iteration Process of n-Gram	**Accuracy-Stats**	**Kernel**
one, two	16.60	polynomial
one, two	16.60	sigmoid
one, two	16.60	rbf

7.2. Linear Support Vector Classification

When both methods use a linear kernel, linear SVC (Linear Support Vector Classification) provides more flexibility in terms of loss and penalty than linear SVM. Larger sample sizes are advantageous for Linear SVC performance, as was already mentioned. Interestingly, the maximum accuracy rate of 71% was obtained using a linear combination of Bigram and Unigram.

Table 11. Best Case Summary Step 7

Percentage-of-Accuracy	69.98
Algorithm-Employed	Linear Support Vector Classification (Linear SVC) with the inclusion of both Unigram and Bigram representations.
Training Duration (in seconds)	0.3867 Second
Testing Duration (in seconds)	0.004 Second

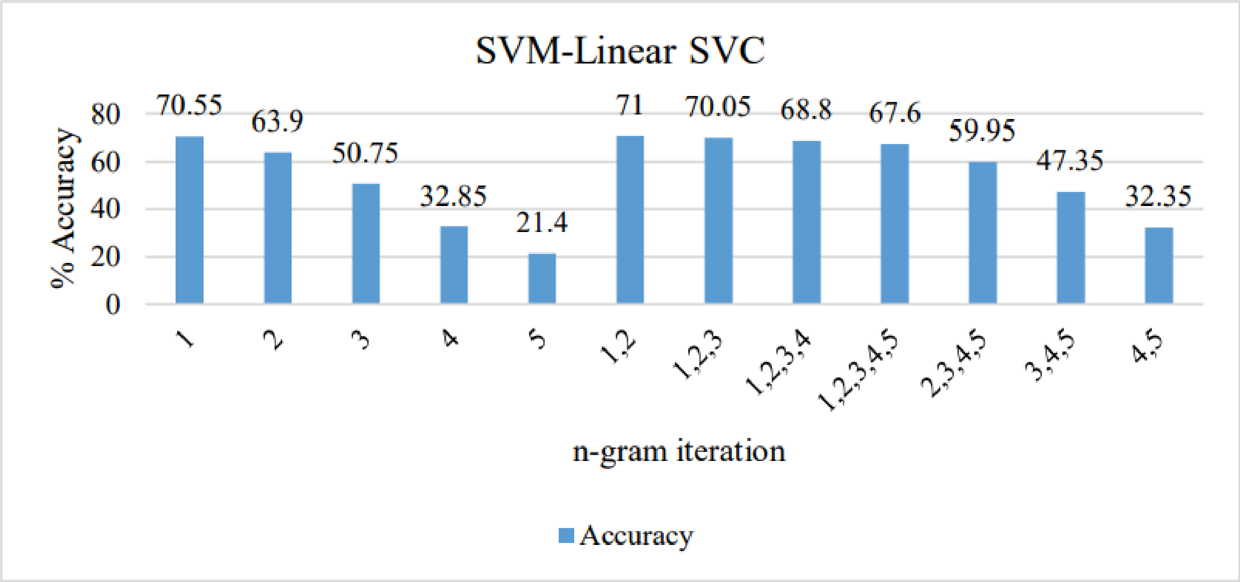


Figure 7. Impact of N-gram on Linear SVC

As shown by table 11, it takes a lot less time than the use cases that have been examined before. This is the highest accuracy rate I could discover after all the testing. As a result, I applied the linear SVC technique using the uni- Gram and bi- Gram combinations on the Environmental change data set.

7.3. Linear Support Vector Classification

Ten thousand tweets on climate change comprised the dataset used in this experiment. The 8,000 randomly chosen tweets from the dataset were utilized as the test dataset and the remaining 2k tweets as the training set. The results shown below are the best results from three independent runs.

On the climate change dataset, linear SVC was used. Bigram and linear SVC together produced an accuracy rate of 61.67%. Figures 8 and Tables 11 and 12 clearly show that, when it comes to datasets particularly designed to address climate change, the training accuracy tends to be greater with conventional datasets. This disparity could result from the fact that, although a climate change dataset contains a smaller bag of words due to its tighter aim, a generic Twitter dataset obtained from many sources offers a wider choice of vocabulary for bag-of-words implementations.

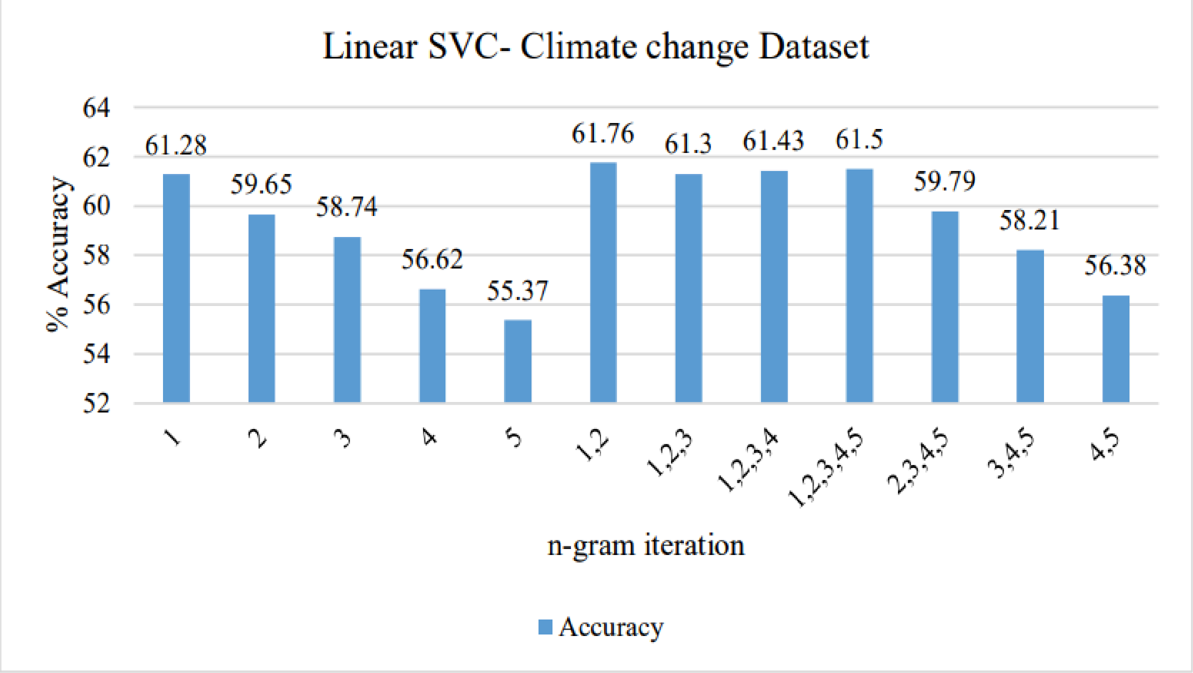


Figure 8: Using Climate Change Specific Dataset with Linear SVC

Table 12. Best Case Summary Step 8

Percentage-of-Accuracy	61.77
Algorithm-Employed	Linear Support Vector Classification (Linear SVC) with the incorporation of both Unigram and Bigram features.
Training Duration (in seconds)	.145 Second
Testing Duration (in seconds)	.002 Second

8. SUMMARY OF TEST CASES

When we look at the impact of bigrams, unigrams, stop words, and halt phrases while working with the Naïve Bayes classifier. The most promising results came from combining stop words with unigrams. Additionally, when the smoothing parameter alpha was added and Multinomial classification using naive-Bayes was used in a number of iterations with various n-gram combinations, it performed better in terms of accuracy than the standard Naïve Bayes classifier. Using a combination of unigrams and bigrams, the MGiven an alpha value of 0.04, the Multi-nomial Naive Bayesian algorithm produced an accuracy of 66.4%.

I then turned to Support Vector Classification (SVC) and played around with other kinds of algorithms, such as Linear SVC. The most successful of them was Linear SVC, which attained an accuracy rate of 71% across a number of n-gram repetitions.

In light of these results, as shown in Table 11, Linear SVC will be the approach of choice going forward for analyzing perspectives on global warming over time.

9. TREND OF GLOBAL WARMING OVER THE YEARS

Compared to previous testing, the accuracy of the Linear SVC technique has been much higher. Using Linear SVC, I looked at how people's opinions regarding global warming evolved over time. Every year, I collect 5,000 tweets and categorize them using the The Support Vector Machine with Linear Support Vector Classification (SVM-Linear SVC) method, and here are the findings derived from this approach:

Table 13. Global Warming Sentiment Analysis per year

	2017	**2016**	**2015**	**2014**	**2013**	**2012**	**2011**	**2010**	**2009**	**2008**	**2007**
Positives	2656	2952	2634	3435	2632	2642	2943	2283	2264	636	31
Negatives	2037	1777	2007	1129	2207	2157	1844	2383	2297	389	15
Neutrals	310	273	359	435	170	198	217	332	436	14	10

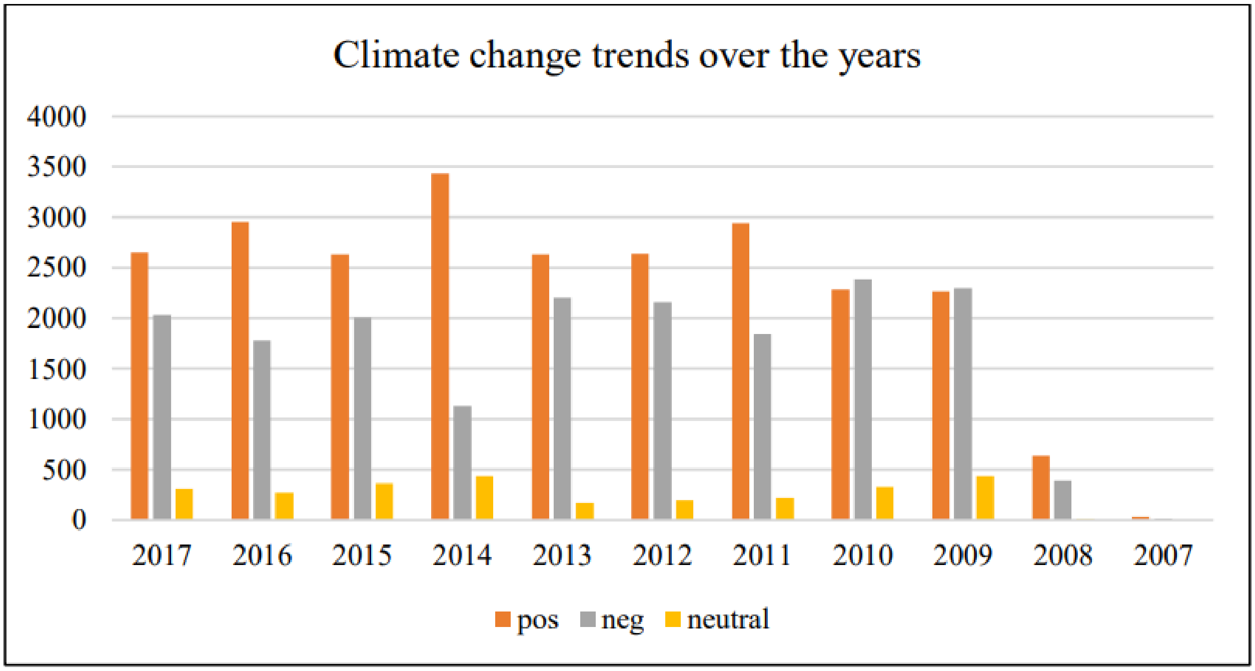


Figure 9. Climate Change Trends over the Years

Above information shows that in 2007, there were just 48 tweets that addressed global warming. But since then, the number of tweets on the subject has steadily increased, suggesting a developing trend. The proportion of respondents who stated they thought global warming was real before 2011 was about the same as the percentage who said they didn't. This tendency did, however, noticeably change in 2011 as more individuals came to accept the existence of global warming.

10. CONCLUSION AND FUTURE WORK

This research used Twitter data collected over a ten-year period to do sentiment analysis on global warming. For classification, methods such as stop word removal, n-gram iterations, additive smoothing, and TF-IDF were used.A number of different models for classification were used, such as the multinomial support vector classification, Support-vector-machine-models (SVM), the naive Bayesian classification algorithm, and linear SVM. When Bigram and linear SVC were combined with unigram, the accuracy was at its maximum, 71%. As a result, Linear SVC was selected to examine tweets from the last ten years on global warming. Sentiment analysis was performed using the Linear SVC algorithm, which collects 5,000 tweets every year. The findings showed a growing pattern of global warming-related tweets since 2008. The proportion of tweets in 2009 and 2010 that said global warming was a hoax was almost equal to that of tweets that said it was true. Nonetheless, a change in pattern was seen, with a greater percentage of good tweets than negative ones in 2014. After 2014, the proportion of positive tweets remained larger than that of negative ones, although it was on the decline. In general, the evidence indicates that a greater proportion of participants are persuaded of the veracity of global warming. Future studies will investigate textual sentiment analysis using deep learning methods like LSTM networks.

REFERENCES

1. Earth Science Communication Team, NASA, Global Climate Change Vital Signs of the Planet https://climate.nasa.gov/effects/
   Retrieved on 07/21/2017
2. Melissa Denchak , March 2016, Are the Effects of Global Warming really that Bad
   https://www.nrdc.org/stories/are-effects-global-warming-really-bad
   Retrieved on 07/21/2017
3. Union of Concerned Scientists, Global Warming Impacts

   http://www.ucsusa.org/our-work/global-warming/science-and-impacts/global-warming-impacts#.WhjTd0qnHIU
   Retrieved on 07/21/2017
4. Joshua Roesslein, Tweepy Documentation
   http://tweepy.readthedocs.io/en/v3.5.0/
   Retrieved on 07/19/2017
5. NLTK 3.2.5 Documentation,
   http://www.nltk.org/_modules/nltk/classify/naivebayes.html
   Retrieved on 08/02/2017
6. Adil Moujahid, July 2014, An Introduction to Text Mining using Twitter Streaming API and Python
   http://adilmoujahid.com/posts/2014/07/twitter-analytics/
   Retrieved on 07/19/2017
7. http://www.tfidf.com/
   Retrieved on
   08/28/2017
8. Mike Waldron, June 2015, Naïve Bayes for Dummies, A Simple Explanation
   http://blog.aylien.com/naive-bayes-for-dummies-a-simple-explanation/
   Retrieved on 07/30/2017
9. Hatem Faheem, July 2015, How are N-grams Used in Machine Learning
   https://www.quora.com/How-are-N-grams-used-in-machine-learning

Retrieved on 08/01/2017

10. Why is N-gram Used in Text Language Identification Instead of Words
https://stats.stackexchange.com/questions/144900/why-is-n-gram-used-in-text-language-identification-instead-of-words
Retrieved on 08/01/2017

11. Abinash Tripaty, Ankit Agarwal, Santanu Kumar Rath, March 2016, Classification of Sentiment Reviews using N-gram Machine Learning Approach
http://www.sciencedirect.com/science/article/pii/S095741741630118X
Retrieved on 07/21/2017

12. Johannes Furnkranz, A Study Using n-gram Features for Text Categorization
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.49.133&rep=rep1&type=pdf
Retrieved on 08/20/2017

13. Manoj Bisht, July 2016, Document Classification using Multinomial Naïve Bayes Classifier
https://www.3pillarglobal.com/insights/document-classification-using-multinomial-naive-bayes-classifier
Retrieved on 09/21/2017

14. Difference between Naïve Bayes and Multinomial Naïve Bayes
https://stats.stackexchange.com/questions/33185/difference-between-naive-bayes-multinomial-naive-bayes
Retrieved on 09/21/2017

15. Difference between Naïve Bayes and Multinomial Naïve Bayes
https://stats.stackexchange.com/questions/33185/difference-between-naive-bayes-multinomial-naive-bayes
Retrieved on 09/21/2017

16. Chistopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, April 2009, Naïve Bayes Text Classification
https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html
Retrieved on 07/20/2017

17. Multinomial Naïve Bayes Classifier

http://scikitlearn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB
Retrieved on 09/25/2017

18. Introduction to Support Vector Machines
http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
Retrieved on 09/25/2017
19. Jacob Perkins, May 2010, Text Classification for Sentiment Analysis-Naïve Bayes Classifier
https://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/
Retrieved on 08/20/2017
20. Sunil Ray, September 2017, Understanding Support Vector Machine Algorithm from Examples
https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
Retrieved on 09/26/2017
21. Sunil Ray, September 2017, 6 Easy Steps to Learn Naïve Bayes Algorithm
https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
Retrieved on 08/20/2017
22. A Beginner's Guide to Recurrent Networks and
LSTMs https://deeplearning4j.org/lstm.html
Retrieved on 08/25/2017
23. Nick Diakopoulos and Shamma, D.A, 2008 US Election debate, Twitter sentiment dataset -
Retrieved on 07/20/2017
24. Niek Sanders, Twitter sentiment corpus
Retrieved on 07/25/2017
25. A lot of sentiment datasets via CS Dept, Cornell
University Retrieved on 07/27/2017
26. Tweepy/Examples/ streaming.py
https://github.com/tweepy/tweepy/blob/master/examples/streaming.py
Retrieved on 07/27/2017

27. Rahul Saxena, February 2017, How the Naïve Bayes Classifier Works in Machine Learning,
http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/
Retrieved on 07/22/2017

28. Scikit learn- Support Vector Machines
http://scikit-learn.org/stable/modules/svm.html
Retrieved on 09/20/2017

29. Jacob Perkins, May 2010, Text Classification for Sentiment analysis- Stopwords and Collocations,
https://streamhacker.com/2010/05/24/text-classification-sentiment-analysis-stopwords-collocations/
Retrieved on 09/05/2017

30. Scikit learn- sklearn.naive_bayes.MultinomialNB
http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
Retrieved on 08/20/2017

31. Ravikiran Janardhana, May 2012, How to Build a Twitter Sentiment Analyzer
https://www.ravikiranj.net/posts/2012/code/how-build-twitter-sentiment-analyzer/
Retrieved on 08/15 /2017

32. Martin Porter, January 2006, The Porter Stemming Algorithm,
https://tartarus.org/martin/PorterStemmer/
Retrieved on 08/20 /2017