TWITTER SENTIMENT ANALYSIS OF MULTIPLE LANGUAGES

A Project Report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Aastha Dubey (201550001)
Nitya Agarwal (201550095)
Anushka Saxena (201550034)
Parth Goyal (201550096)
Rahul Singh (201550112)

Under the Guidance of
Mr. Rahul Pradhan, Assistant Professor
Department of Computer Engineering & Applications
Institute of Engineering & Technology

GLA University
Mathura- 281406, INDIA
October, 2023

Department of Computer Engineering

GLA University, Mathura , Uttar Pradesh 281406

CERTIFICATE

It is certified that the contents and form of thesis entitled **"Twitter Sentiment Analysis"** submitted by *Anushka Saxena , Aastha Dubey , Nitya Agrawal , Parth Goyal , Rahul* Kumar to the Ganeshi Lal Agarwal University, Mathura, in partial fulfillment for the award of the degree of B. Tech in (Computer Science) is a record of project work carried out by us under our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

Mentor : ______________________________

(Mr. Rahul Pradhan)

ACKNOWLEDGEMENTS

We are deeply thankful to our mentor , Mr. Rahul Pradhan Sir for helping us throughout the course in accomplishing our final project. Their guidance, support and motivation enabled us in achieving the objectives of the project.

ABSTRACT

In the age of digital communication and social media, Twitter stands as a global platform that encapsulates diverse voices and opinions. While it fosters meaningful conversations and connections, it also serves as a breeding ground for offensive language and harmful content. Understanding the prevalence and sentiment associated with foul language in tweets is of paramount importance in shaping a healthier and more respectful online environment.

This major project embarks on an ambitious journey to analyse and interpret Twitter data containing foul language in a multilingual context. By examining tweets in languages including English, Hindi, Bengali, and others, we aim to unravel the intricate tapestry of sentiments woven into these expressions. The project strives to delve deep into the dynamics of online discourse, offering valuable insights into the emotional undercurrents that accompany offensive language across linguistic boundaries.

TABLE OF CONTENTS

Chapter 1
INTRODUCTION

In the rapidly evolving landscape of social media, Twitter stands as a dynamic and influential platform where individuals, organizations, and communities converge to share their thoughts, opinions, and experiences. While Twitter has been instrumental in connecting people worldwide and fostering dialogue, it is no secret that it has also become a fertile ground for the dissemination of offensive language, hate speech, andderogatory remarks. This major project embarks on a comprehensive exploration of the phenomenon of foul language on Twitter, seeking to unravel the complexities and subtleties of this pervasive issue in a multilingual context.

The project is driven by a deep-seated concern for the quality of online discourse and the well-being of social media users. Foul language, hate speech, and offensive content can have far-reaching consequences, ranging from the degradation of online communities to the perpetuation of stereotypes and prejudices. Understanding the prevalence, sentiment, and linguistic nuances of offensive language across different languages is essential for developing effective strategies to mitigate its impact.

1.1 MOTIVATION

The motivation behind this project stems from the critical need to foster a respectful and inclusive online environment. As the internet continues to serve as a primary medium for communication and information dissemination, it is imperative that platforms like Twitter actively address the challenges posed by offensive language. By gaining a deep insight into the nature of foul language and its sentiment in multilingual tweets, we aim to contribute to the creation of safer and more enjoyable online spaces for users of diverse linguistic backgrounds.

1.2 Scope and Significance

This project's scope encompasses the collection, analysis, and interpretation of tweets containing foul language in multiple languages, including but not limited to English, Hindi, Bengali, and others. By examining both the content and sentiment associated with offensive language, we seek to achieve the following objectives:

- Develop a comprehensive dataset of tweets that contain foul language in various languages.
- Utilize state-of-the-art natural language processing (NLP) techniques and machine learning models to conduct sentiment analysis on these tweets.
- Investigate language-specific nuances and variations in the use of offensive language.

- Identify common offensive terms and quantify their prevalence across languages.
- Visualize and present the findings in a manner that facilitates a nuanced understanding of the issue.

The significance of this project extends beyond the confines of Twitter. It has implications for content moderation, user experience enhancement, linguistics research, and the broader discourse on responsible online communication. By addressing the challenges posed by foul language in tweets, we aim to contribute to the creation of a more harmonious and respectful digital ecosystem.

	Machine says yes	**Machine says no**
Human says yes	**tp**	**fn**
Human says no	**fp**	**tn**

Table 1: A Typical 2x2 Confusion Matrix

Precision(P) = $\frac{tp}{tp+fp}$ **Recall(R) =** $\frac{tp}{tp+fn}$ **Accuracy(A) =** $\frac{tp+tn}{tp+tn+f+fp+fn}$

F1 = $\frac{2.P.R}{P+R}$ **True Rate(T) =** $\frac{tp}{tp+fn}$ **False-alarm Rate(F) =** $\frac{fp}{tp+fn}$

CHAPTER 2
PROJECT OBJECTIVES

2.1 REQUIREMENT ANALYSIS

2.1.1 SOFTWARE COMPONENTS

- Windows Operating System
- Google Colab
- Python 3.10.4
- Jupyter Lab
- Jupyter Notebook

2.1.2 HARDWARE COMPONENTS

- Laptops / Computers with Minimum 8 GB RAM

2.2 MODULES AND FUNCTIONALITIES

The core objectives of this project encompass:

Robust Data Collection: Curate a substantial and diverse dataset of tweets containing foul language in multiple languages, ensuring representation from various geographical regions and cultures.

Data Preprocessing: Employ cutting-edge natural language processing (NLP) techniques to preprocess and clean the data, making it amenable for in-depth analysis.

Sentiment Analysis: Utilize state-of-the-art machine learning models and sentiment analysis tools to decipher the overall sentiment (positive, negative, neutral) associated with tweets containing foul language.

Cross-Linguistic Comparison: Explore and analyze the nuances of sentiment across languages, identifying potential variations in the perception of offensive language.

Offensive Term Identification: Unearth common offensive terms and quantify their prevalence in each language, shedding light on the linguistic diversity of online profanity.

Data Visualization: Create compelling visualizations such as charts, graphs, and heatmaps to present the findings in an accessible and informative manner.

Grammatical features (like "Parts of Speech Tagging" or POS tagging) are also commonly used in this domain. The concept is to tag each word of the tweet in terms of what part of speech it belongs to: noun, pronoun, verb, adjective, adverb, interjections, intensifiers etc. The concept is to detect patterns based on these POS and use them in the classification process. For example it has been reported that objective tweets contain more common nouns and third-person verbs than subjective tweets [3], so if a tweet to be classified has a proportionally large usage of common nouns and verbs in third person, that tweet would have a greater probability of being objective (according to this particular feature). Similarly subjective tweets contain more adverbs, adjectives and interjections [3]. These relationships are demonstrated in the figures below:

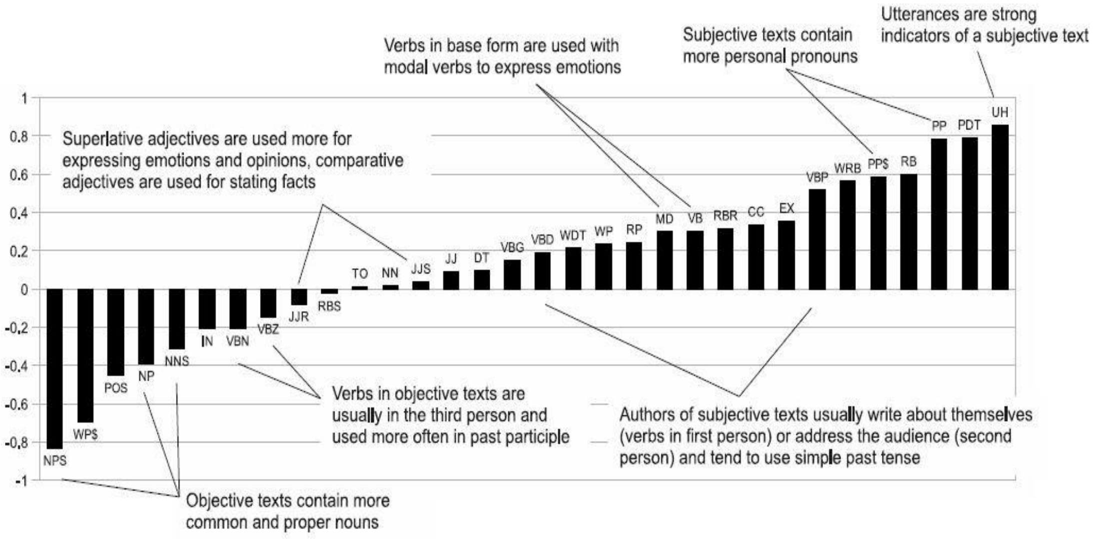

Figure 1: Using POS Tagging as features for objectivity/subjectivity classification

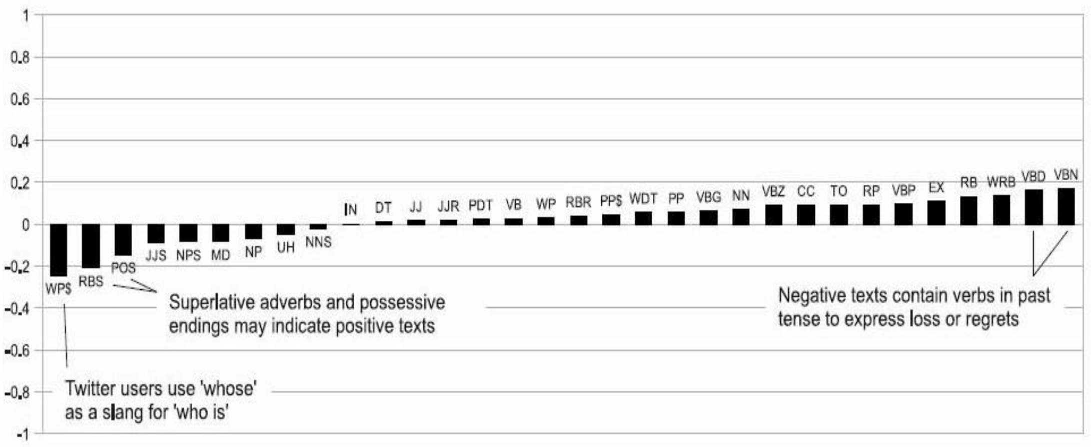

Figure 2: Using POS Tagging as features in positive/negative classification

However there is still conflict whether Parts-of-Speech are a useful feature for sentiment classification or not. Some researchers argue in favour of good POS features (*e.g.*, [10]) while others not recommending them (*e.g.*, [7]).

Besides from these much work has been done in exploring a class of features pertinent only to micro blogging domain. Presence of URL and number of capitalized words/alphabets in a tweet have been explored by Koulompis et al. [7] and Barbosa et al. [10]. Koulmpis also reports positive results for using emoticons and internet slang words as features. Brody et al. does study on word lengthening as a sign of subjectivity in a tweet [13]. The paper reports positive results for their study that the more number of cases a word has of lengthening, the more chance there of that word being a strong indication of subjectivity.

The most commonly used classification techniques are the Naive Bayes Classifier and State Vector Machines. Some researchers like Barbosa et al. publish better results for SVMs [10] while others like Pak et al. support Naive Bayes [3]. (1-9) and (2-6) also report good results for Maximum Entropy classifier.

It has been observed that having a larger training sample pays off to a certain degree, after which the accuracy of the classifier stays almost constant even if we keep adding more labelled tweets in the training data [10]. Barbosa et al. used tweets labelled by internet resources (*e.g.*, [28]), instead of labelling them by hand, for training the classifier. Although there is loss of accuracy of the labelled samples in doing so (which is modelled as increase in noise) but it has been observed that if the accuracy of training labels is greater than 50%, the more the labels, the higher the accuracy of the resulting classifier. So in this way if there are an extremely large number of tweets, the fact that our labels are noisy and inaccurate can be compensated for [10]. On the other hand Pak et al. and Go et al. [2] use presence of positive or negative emoticons to assign labels to the tweets [3]. Like in the above case they used large number of tweets to reduce effect of noise in their training data.

Some of the earliest work in this field classified text only as positive or negative, assuming that all the data provided is subjective (for example in [2] and [5]). While this is a good assumption for something like movie reviews but when analyzing tweets and blogs there is a lot of objective text we have to consider, so incorporating neutral class into the classification process is now becoming a norm. Some of the work which has included neutral class into their classification process includes [7], [10], [3] and [16].

There has also been very recent research of classifying tweets according to the mood expressed in them, which goes one step further. Bollen et al. explores this area and develops a technique to classify tweets into six distinct moods: tension, depression, anger, vigour, fatigue and confusion [9]. They use an extended version of Profile of Mood States (POMS): a widely accepted psychometric instrument. They generate a word dictionary and assign them weights corresponding to each of the six mood states, and then they represented each tweet as a vector corresponding to these six dimensions. However not much detail has been provided into how they built their customized lexicon and what technique did they use for classification.

CHAPTER 3
FUNCTIONALITY AND DESIGN

The process of designing a functional classifier for sentiment analysis can be broken down into five basic categories. They are as follows:

I. Data Acquisition
II. Human Labelling
III. Feature Extraction
IV. Classification
V. TweetMood Web Application

Data Acquisition:

Data in the form of raw tweets is acquired by using the python library "tweestream" which provides a package for simple twitter streaming API [26]. This API allows two modes of accessing tweets: SampleStream and FilterStream. SampleStream simply delivers a small, random sample of all the tweets streaming at a real time. FilterStream delivers tweet which match a certain criteria. It can filter the delivered tweets according to three criteria:

- Specific keyword(s) to track/search for in the tweets
- Specific Twitter user(s) according to their user-id's
- Tweets originating from specific location(s) (only for geo-tagged tweets).

A programmer can specify any single one of these filtering criteria or a multiple combination of these. But for our purpose we have no such restriction and will thus stick to the SampleStream mode.

Since we wanted to increase the generality of our data, we acquired it in portions at different points of time instead of acquiring all of it at one go. If we used the latter approach then the generality of the tweets might have been compromised since a significant portion of the tweets would be referring to some certain trending topic and would thus have more or less of the same general mood or sentiment. This phenomenon has been observed when we were going through our sample of acquired tweets. For example the sample acquired near Christmas and New Year’s had a significant portion of tweets referring to these joyous events and were thus of a generally positive sentiment. Sampling our data in portions at different points in time would thus try to minimize this problem. Thus forth, we acquired data at four different points which would be 17th of December 2011, 29th of December 2011, 19th of January 2012 and 8th of February 2012.

A tweet acquired by this method has a lot of raw information in it which we may or may not find useful for our particular application. It comes in the form of the python “dictionary” data type with various key-value pairs. A list of some key-value pairs are given below:

- Whether a tweet has been favourited
- User ID
- Screen name of the user
- Original Text of the tweet
- Presence of hashtags
- Whether it is a re-tweet
- Language under which the twitter user has registered their account
- Geo-tag location of the tweet
- Date and time when the tweet was created

Since this is a lot of information we only filter out the information that we need and discard the rest. For our particular application we iterate through all the tweets in our sample and save the actual text content of the tweets in a separate file given that language of the twitter is user's account is specified to be English. The original text content of the tweet is given under the dictionary key "**text**" and the language of user's account is given under "**lang**".

Since human labelling is an expensive process we further filter out the tweets to be labelled so that we have the greatest amount of variation in tweets without the loss of generality. The filtering criteria applied are stated below:

- Remove Retweets (any tweet which contains the string "RT")
- Remove very short tweets (tweet with length less than 20 characters)
- Remove non-English tweets (by comparing the words of the tweets with a list of 2,000 common English words, tweets with less than 15% of content matching threshold are discarded)
- Remove similar tweets (by comparing every tweet with every other tweet, tweets with more than 90% of content matching with some other tweet is discarded)

After this filtering roughly 30% of tweets remain for human labelling on average per sample, which made a total of 10,173 tweets to be labelled.

Human Labelling:

For the purpose of human labelling we made three copies of the tweets so that they can be labelled by four individual sources. This is done so that we can take average opinion of people on the sentiment of the tweet and in this way the noise and inaccuracies in labelling can be minimized. Generally speaking the more copies of labels we can get the better it is, but we have to keep the cost of labelling in our mind, hence we reached at the reasonable figure of three.

We labelled the tweets in four classes according to sentiments expressed/observed in the tweets: positive, negative, neutral/objective and ambiguous. We gave the following guidelines to our labellers to help them in the labelling process:

- **Positive**: If the entire tweet has a positive/happy/excited/joyful attitude or if something is mentioned with positive connotations. Also if more than one sentiment is expressed in the tweet but the positive sentiment is more dominant. Example: "*4 more years of being in shithole Australia then I move to the USA! :D*".
- **Negative**: If the entire tweet has a negative/sad/displeased attitude or if something is mentioned with negative connotations. Also if more than one sentiment is expressed in the tweet but the negative sentiment is more dominant. Example: "*I want an android now this iPhone is boring :S*".
- **Neutral/Objective**: If the creator of tweet expresses no personal sentiment/opinion in the tweet and merely transmits information. Advertisements of different products would be labelled under this category. Example: "*US House Speaker vows to stop Obama contraceptive rule... http://t.co/cyEWqKlE*".
- **Ambiguous**: If more than one sentiment is expressed in the tweet which are equally potent with no one particular sentiment standing out and becoming more obvious. Also if it is obvious that some personal opinion is being expressed here but due to lack of reference to context it is difficult/impossible to accurately decipher the sentiment expressed. Example: "*I kind of like heroes and don't like it at the same time...*". Finally if the context of the tweet is not apparent from the information available. Example: "*That's exactly how I feel about avengers haha*".
- **<Blank>**: Leave the tweet unlabelled if it belongs to some language other than English so that it is ignored in the training data.

Besides this labellers were instructed to keep personal biases out of labelling and make no assumptions, i.e. judge the tweet not from any past extra personal information and only from the information provided in the current individual tweet.

Once we had labels from four sources our next step was to combine opinions of three people to get an averaged opinion. The way we did this is through majority vote.

So for example if a particular tweet had to two labels in agreement, we would label the overall tweet as such. But if all three labels were different, we labelled the tweet as “unable to reach a majority vote”. We arrived at the following statistics for each class after going through majority voting.

- Positive: 2543 tweets
- Negative: 1877 tweets
- Neutral: 4543 tweets
- Ambiguous: 451 tweets
- Unable to reach majority vote: 390 tweets
- Unlabelled non-English tweets: 369 tweets

So if we include only those tweets for which we have been able to achieve a positive, negative or neutral majority vote, we are left with 8963 tweets for our training set. Out of these 4543 are objective tweets and 4420 are subjective tweets (sum of positive and negative tweets).

We also calculated the human-human agreement for our tweet labelling task, results of which are as follows:

	Human 1: Human 2	Human 2: Human 3	Human 1: Human 3
Strict	58.9%	59.9%	62.5%
Lenient	65.1%	67.1%	73.0%

Table 4: Human-Human Agreement in Tweet Labelling

In the above matrix the “strict” measure of agreement is where all the label assigned by both human beings should match exactly in all cases, while the “lenient” measure is in which if one person marked the tweet as “ambiguous” and the other marked it as something else, then this would not count as a disagreement. So in case of the "lenient" measure, the ambiguous class could map to any other class. So since the human-human agreement lies in the range of 60-70% (depending upon our definition of agreement), this shows us that sentiment classification is inherently a difficult task even for human beings. We will now look at another table presented by Kim et al. which shows human-human agreement in case labelling individual adjectives and verbs. [14]

	Adjectives	Verbs
	Human 1: Human 2	Human 1: Human 3
Strict	76.19%	62.35%
Lenient	88.96%	85.06%

Table 5: Human- Human Agreement in Verbs / Adjectives Labelling [6]

Over here the strict measure is when classification is between the three categories of positive, negative and neutral, while the lenient measure the positive and negative classes into one class, so now humans are only classifying between neutral and subjective classes. These results reiterate our initial claim that sentiment analysis is an inherently difficult task. These results are higher than our agreement results because in this case humans are being asked to label individual words which is an easier task than labelling entire tweets.

Feature Extraction:

Now that we have arrived at our training set we need to extract useful features from it which can be used in the process of classification. But first we will discuss some text formatting techniques which will aid us in feature extraction:

- Tokenization: It is the process of breaking a stream of text up into words, symbols and other meaningful elements called “tokens”. Tokens can be separated by whitespace characters and/or punctuation characters. It is done so that we can look at tokens as individual components that make up a tweet [19].
- Url’s and user references (identified by tokens “http” and “@”) are removed if we are interested in only analyzing the text of the tweet.
- Punctuation marks and digits/numerals may be removed if for example we wish to compare the tweet to a list of English words.
- Lowercase Conversion: Tweet may be normalized by converting it to lowercase which makes it’s comparison with an English dictionary easier.
- Stemming: It is the text normalizing process of reducing a derived word to its root or stem [28]. For example a stemmer would reduce the phrases “stemmer”, “stemmed”, “stemming” to the root word “stem”. Advantage of stemming is that it makes comparison between words simpler, as we do not need to deal with complex grammatical transformations of the word. In our case we employed the algorithm of “porter stemming” on both the tweets and the dictionary, whenever there was a need of comparison.
- Stop-words removal: Stop words are class of some extremely common words which hold no additional information when used in a text and are thus claimed to be useless [19]. Examples include “a”, “an”, “the”, “he”, “she”, “by”, “on”, etc. It is sometimes convenient to remove these words because they hold no additional information since they are used almost equally in all classes of text, for example when computing prior-sentiment-polarity of words in a tweet according to their frequency of occurrence in different classes and using this

polarity to calculate the average sentiment of the tweet over the set of words used in that tweet.

- Parts-of-Speech Tagging: POS-Tagging is the process of assigning a tag to each word in the sentence as to which grammatical part of speech that word belongs to, i.e. noun, verb, adjective, adverb, coordinating conjunction etc.

Now that we have discussed some of the text formatting techniques employed by us, we will move to the list of features that we have explored. As we will see below a feature is any variable which can help our classifier in differentiating between the different classes. There are two kinds of classification in our system (as will be discussed in detail in the next section), the objectivity / subjectivity classification and the positivity / negativity classification. As the name suggests the former is for differentiating between objective and subjective classes while the latter is for differentiating between positive and negative classes.

The list of features explored for objective / subjective classification is as below:

- Number of exclamation marks in a tweet
- Number of question marks in a tweet
- Presence of exclamation marks in a tweet
- Presence of question marks in a tweet
- Presence of url in a tweet
- Presence of emoticons in a tweet
- Unigram word models calculated using Naive Bayes
- Prior polarity of words through online lexicon MPQA
- Number of digits in a tweet
- Number of capitalized words in a tweet
- Number of capitalized characters in a tweet
- Number of punctuation marks / symbols in a tweet

- Ratio of non-dictionary words to the total number of words in the tweet
- Length of the tweet
- Number of adjectives in a tweet
- Number of comparative adjectives in a tweet
- Number of superlative adjectives in a tweet
- Number of base-form verbs in a tweet
- Number of past tense verbs in a tweet
- Number of present participle verbs in a tweet
- Number of past participle verbs in a tweet
- Number of 3rd person singular present verbs in a tweet
- Number of non-3rd person singular present verbs in a tweet
- Number of adverbs in a tweet
- Number of personal pronouns in a tweet
- Number of possessive pronouns in a tweet
- Number of singular proper noun in a tweet
- Number of plural proper noun in a tweet
- Number of cardinal numbers in a tweet
- Number of possessive endings in a tweet
- Number of wh-pronouns in a tweet
- Number of adjectives of all forms in a tweet
- Number of verbs of all forms in a tweet
- Number of nouns of all forms in a tweet
- Number of pronouns of all forms in a tweet

The list of features explored for positive / negative classification are given below:

- Overall emoticon score (where 1 is added to the score in case of positive emoticon, and 1 is subtracted in case of negative emoticon)

- Overall score from online polarity lexicon MPQA (where presence of strong positive word in the tweet increases the score by 1.0 and the presence of weak negative word would decrease the score by 0.5)
- Unigram word models calculated using Naive Bayes
- Number of total emoticons in the tweet
- Number of positive emoticons in a tweet
- Number of negative emoticons in a tweet
- Number of positive words from MPQA lexicon in tweet
- Number of negative words from MPQA lexicon in tweet
- Number of base-form verbs in a tweet
- Number of past tense verbs in a tweet
- Number of present participle verbs in a tweet
- Number of past participle verbs in a tweet
- Number of 3rd person singular present verbs in a tweet
- Number of non-3rd person singular present verbs in a tweet
- Number of plural nouns in a tweet
- Number of singular proper nouns in a tweet
- Number of cardinal numbers in a tweet
- Number of prepositions or coordinating conjunctions in a tweet
- Number of adverbs in a tweet
- Number of wh-adverbs in a tweet
- Number of verbs of all forms in a tweet

Next we will give mathematical reasoning of how we calculate the unigram word models using Naive Bayes. The basic concept is to calculate the probability of a word belonging to any of the possible classes from our training sample. Using mathematical formulae we will demonstrate an example of calculating probability of word belong to objective and subjective class. Similar steps would need to be taken for positive and negative classes as well.

We will start by calculating the probability of a word in our training data for belonging to a particular class:

$$P(word_1|obj) = \frac{count(word_1\ in\ obj\ class)}{count(total\ words\ in\ obj)}$$

We now state the Bayes' rule [19]. According to this rule, if we need to find the probability of whether a tweet is objective, we need to calculate the probability of tweet given the objective class and the prior probability of objective class. The term *P(tweet)* can be substituted with *P(tweet | obj) + P(tweet | subj).*

$$P(obj|tweet) = \frac{P(tweet|obj).P(obj)}{P(tweet)}$$

Now if we assume independence of the unigrams inside the tweet (i.e. the occurrence of a word in a tweet will not affect the probability of occurrence of any other word in the tweet) we can approximate the probability of tweet given the objective class to a mere product of the probability of all the words in the tweet belonging to objective class. Moreover, if we assume equal class sizes for both objective and subjective class we can ignore the prior probability of the objective class. Henceforth we are left with the following formula, in which there are two distinct terms and both of them are easily calculated through the formula mention above.

$$P(obj|tweet) = \frac{\prod_{i=1}^{N}\ [P(word_i|obj)}{\prod_{i=1}^{N}\ [P(word_i|obj) + \prod_{i=1}^{N}\ [P(word_i|subj)}$$

Now that we have the probability of objectivity given a particular tweet, we can easily calculate the probability of subjectivity given that same tweet by simply subtracting the earlier term from 1. This is because probabilities must always add to 1. So if we have information of *P(obj | tweet)* we automatically know *P(subj | tweet).*

$$\boldsymbol{P(subj|tweet) = 1 - P(obj|tweet)}$$

Finally we calculate P(obj | tweet) for every tweet and use this term as a single feature in our objectivity / subjectivity classification.

There are two main potential problems with this approach. First being that if we include every unique word used in the data set then the list of words will be too large making the computation too expensive and time-consuming. To solve this we only include words which have been used at least 5 times in our data. This reduces the size of our dictionary for objective / subjective classification from 11,216 to 2,320. While for positive / negative classification unigram dictionary size is reduced from 6,502 to 1,235 words.

The second potential problem is if in our training set a particular word only appears in a certain class only and does not appear at all in the other class (for example if the word is misspelled only once). If we have such a scenario then our classifier will always classify a tweet to that particular class (regardless of any other features present in the tweet) just because of the presence of that single word. This is a very harsh approach and results in over-fitting. To avoid this we make use of the technique known as “Laplace Smoothing”. We replace the formula for calculating the probability of a word belonging to a class with the following formula:

$$\boldsymbol{P(word_1|obj) = \frac{count(word_1\ in\ obj\ class) +\ x}{count(total\ words\ in\ obj) +\ x(total\ unique\ words\ in\ obj}}$$

In this formula “x” is a constant factor called the smoothing factor, which we have arbitrarily selected to be 1. How this works is that even if the count of a word in a particular class is zero, the numerator still has a small value so the probability of a word belonging to some class will never be equal to zero. Instead if the probability would have been zero according to the earlier formula, it would be replace by a very small non-zero probability.

The final issue we have in feature selection is choosing the best features from a large number of features. Our ultimate aim is to achieve the greatest accuracy of our classifier while using least number of features. This is because adding new feature add to the dimensionality of our classification problem and thus add to the complexity of our classifier. This increase in complexity may not necessarily be linear and may even be quadratic so it is preferred to keep the features at a minimum low. Another issue we have with too many features is that our training data may be over-fit and it may confuse the classifier when doing classification on an unknown test set, so the accuracy of the classifier may even decrease. To solve this issue we select the most pertinent features by computing the information-gain of all the features under exploration and then selecting the features with highest information gain. We used WEKA machine learning tool for this task of feature selection [17].

We explored a total of 33 features for objectivity / subjectivity classification and used WEKA to calculate the information gain from each of these features. The resulting graph is shown below:

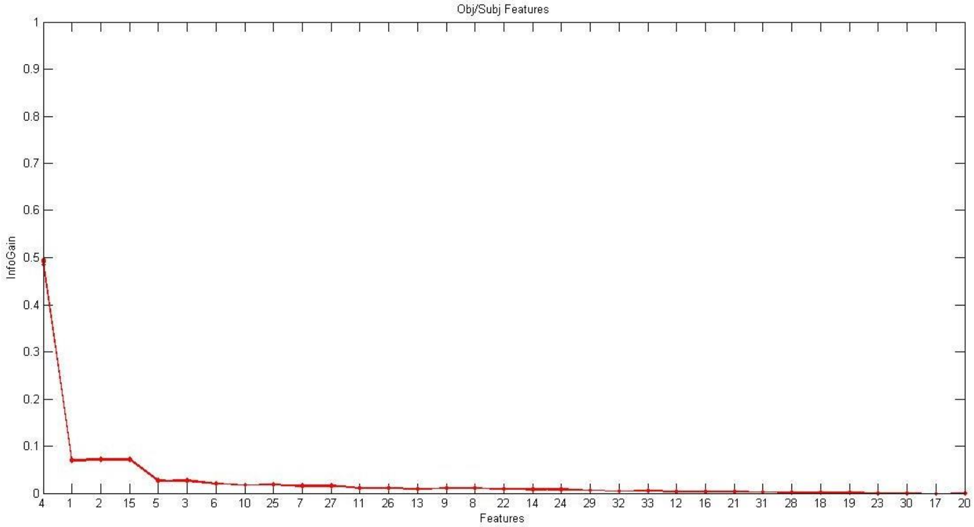

Figure 3: Information Gain of Objectivity / Subjectivity Features

This graph is basically the super-imposition of 10 different graphs, each one arrived through one fold out of the 10-fold cross validation we performed. Since we see that all the graphs are nicely overlapping so the results each fold are almost the same which shows us that the features we select will perform best in all the scenarios. We selected the best 5 features from this graph which are as follows:

1. Unigram word models (for prior probabilities of words belonging to objective / subjective classes)
2. Presence of URL in tweet
3. Presence of emoticons in tweet
4. Number of personal pronouns in tweet
5. Number of exclamation marks in tweet

Similarly we explored 22 features for positive / negative classification and used WEKA to calculate the information gain from each of these features. The resulting graph is shown below:

We also calculated the human-human agreement for our tweet labelling task, results of which are as follows:

	Human 1: Human 2	Human 2: Human 3	Human 1: Human 3
Strict	58.9%	59.9%	62.5%
Lenient	65.1%	67.1%	73.0%

Table 4: Human-Human Agreement in Tweet Labelling

	Adjectives	Verbs
	Human 1: Human 2	Human 1: Human 3
Strict	76.19%	62.35%
Lenient	88.96%	85.06%

Table 5: Human- Human Agreement in Verbs / Adjectives Labelling [6]

Over here the strict measure is when classification is between the three categories of positive, negative and neutral, while the lenient measure the positive and negative classes into one class, so now humans are only classifying between neutral and subjective classes. These results reiterate our initial claim that sentiment analysis is an inherently difficult task. These results are higher than our agreement results because in this case humans are being asked to label individual words which is an easier task than labelling entire tweets.

We will start by calculating the probability of a word in our training data for belonging to a particular class:

$$P(word_1|obj) = \frac{count(word_1\ in\ obj\ class)}{count(total\ words\ in\ obj)}$$

TweetMood Insights Web Application:

We designed a web application which performed real-time sentiment analysis on Twitter on tweets that matched particular keywords provided by the user. For example if a user is interested in performing sentiment analysis on tweets which contain the word “Rude” he / she will enter that keyword and the web application will perform the appropriate sentiment analysis and display the results for the user.

Figure 6: Tweet Mood Insights web-application logo

Chapter 4

IMPLEMENTATION AND RESULT DISCUSSION

4.1 OUTCOME

The expected outcomes of this project encompass:

1. A comprehensive dataset of tweets containing foul language in multiple languages.
2. Sentiment analysis results, depicting the distribution of positive, negative, and neutral sentiments across languages.
3. Comparative analysis of offensive language use, highlighting linguistic and cultural differences.
4. Data visualizations that facilitate a deeper understanding of the findings**.**

We will first present our results for the objective / subjective and positive / negative classifications. These results act as the first step of our classification approach. We only use the short-listed features for both of these results. This means that for the objective / subjective classification we have 5 features and for positive / negative classification we have 3 features. For both of these results we use the Naïve Bayes classification algorithm, because that is the algorithm we are employing in our actual classification approach at the first step. Furthermore all the figures reported are the result of 10-fold cross validation. We take an average of each of the 10 values we get from the cross validation.

Classes	True Positive	False Positive	Recall	Precision	F-measure
Objective	0.73	0.26	0.74	0.73	0.73
Subjective	0.74	0.27	0.725	0.73	0.73
Average	0.73	0.27	0.73	0.73	0.73

Table 6: Results from Objective / Subjective Classification

4.2 SIGNIFICANCE

This project holds significant importance for various stakeholders:

Content Moderation: The insights gained can enhance the effectiveness of content moderation and filtering systems, making Twitter a safer platform.

Linguistic and Sociolinguistic Studies: The project provides valuable data for linguists and sociolinguists studying language use in online environments, shedding light on the evolving nature of language.

User Experience: By fostering a more respectful online community, this project contributes to a positive user experience for Twitter's global audience.

In addition to the above information, we make a condition while reporting the results of polarity classification (which differentiates between positive and negative classes) that only subjective labelled tweets are used to calculate these results. However, in case of final classification approach, any such condition is removed and basically both objectivity and polarity classifications are applied to all tweets regardless of whether they are labelled objective or subjective.

If we compare these results to those provided by Wilson et al. [16] (results are displayed in Table 2 and Table 3 of this report) we see that although the accuracy of neutral class falls from 82.1% to 73% if we use our classification instead of theirs. However, for all other classes we report significantly greater results. Although the results presented by Wilson et al. are not from Twitter data they are of phrase level sentiment analysis which is very close in concept to Twitter sentiment analysis.

Next we will compare our results with those presented by Go et al. [2]. The results presented by this paper are as follows:

Features	Naive Bayes	Max Entropy	SVM
Unigram	81.3%	80.5%	82.2%
Bigram	81.6%	79.1%	78.8%
Unigram + Bigram	82.7%	83.0%	81.6%
Unigram + POS	79.9%	79.9%	81.9%

Table 8: Positive / Negative Classification Results presented by (1-9)

If we compare these results to ours, we see that they are more or less similar. However, we arrive at comparable results with just 10 features and about 9,000 training data. In contrast to this, they used about 1.6 million noisy labels. Their labels were noisy in the sense that the tweets that contained positive emoticons were labelled as positive, while those with negative emoticons were labelled negative. The rest of the tweets (which did not contain any emoticon) were discarded from the data set. So in this way they hoped to achieve high results without human labelling but at the cost of using humongous large number amount of data set.

Next we will present our results for the complete classification. We note that the best results are reached through Support Vector Machine being applied at the second stage of the classification process. Hence the results below will only pertain to those of SVM. These results use a total of two features: P(objectivity | tweet) and P(positivity | tweet). But if we include all the features employed in step 1 of the classification process, we have a list of 8 shortlisted features (3 for polarity classification and 5 for objectivity classification). The following results are reported after conducting 10-fold cross validation:

Classes	True Positive	False Positive	Recall	Precision	F-measure
Objective	0.77	0.27	0.77	0.75	0.76
Positive	0.66	0.11	0.66	0.70	0.68
Negative	0.60	0.10	0.59	0.61	0.60
Average	0.70	0.19	0.703	0.703	0.703

Table 9: Final Results using SVM at Step 2 and Naive Bayes at Step 1

Finally we conclude that our classification approach provides improvement in accuracy by using even the simplest features and small amount of data set. However there are still a number of things we would like to consider as future work which we mention in the next section.

Chapter 5
CONCLUSION

In conclusion, this project has provided valuable insights into the world of offensive language on Twitter, spanning multiple languages and cultures. By conducting sentiment analysis and exploring language-specific nuances, we have laid the groundwork for more effective content moderation, enhanced user experiences, and a deeper understanding of online communication. As we move forward, the knowledge gained from this project will serve as a valuable resource in the ongoing effort to create a safer, more respectful, and inclusive digital ecosystem. It is our hope that the findings presented here contribute to a more harmonious and responsible online environment for users worldwide.

The task of sentiment analysis, especially in the domain of micro-bloging, is still in the developing stage and far from complete. So we propose a couple of ideas which we feel are worth exploring in the future and may result in further improved performance

Right now we have worked with only the very simplest unigram models; we can improve those models by adding extra information like closeness of the word with a negation word. We could specify a window prior to the word (a window could for example be of 2 or 3 words) under consideration and the effect of negation may be incorporated into the model if it lies within that window. The closer the negation word is to the unigram word whose prior polarity is to be calculated, the more it should affect the polarity. For example if the negation is right next to the word, it may simply reverse the polarity of that word and farther the negation is from the word the more minimized ifs effect should be.

One more feature we that is worth exploring is whether the information about relative position of word in a tweet has any effect on the performance of the classifier. Although Pang et al. explored a similar feature and reported negative results, their results were based on reviews which are very different from tweets and they workedon an extremely simple model.

In this research we are focussing on general sentiment analysis. There is potential of work in the field of sentiment analysis with partially known context. For example we noticed that users generally use our website for specific types of keywords which can divided into a couple of distinct classes, namely: politics/politicians, celebrities, products/brands, sports/sportsmen, media/movies/music. So we can attempt to perform separate sentiment analysis on tweets that only belong to one of these classes (i.e. the training data would not be general but specific to one of these categories) and compare the results we get if we apply general sentiment analysis on it instead.

REFERENCES:

[1] Albert Biffet and Eibe Frank. Sentiment Knowledge Discovery in Twitter Streaming Data. *Discovery Science, Lecture Notes in Computer Science*, 2010,
Volume 6332/2010, 1-15, DOI: 10.1007/978-3-642-16184-1_1

[2] Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. *Project Technical Report, Stanford University*, 2009.

[3] Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *In Proceedings of international conference on Language Resources and Evaluation (LREC)*, 2010.

[4] Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner and Isabell M. Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. *In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010. Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP), 2002.

[5] Chenhao Tan, Lilian Lee, Jie Tang, Long Jiang, Ming Zhou and Ping Li. User Level Sentiment Analysis Incorporating Social Networks. *In Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, 2011.

[6] Efthymios Kouloumpis, Theresa Wilson and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! *In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[7] Hatzivassiloglou, V., & McKeown, K.R.. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL*, 2009.

[8] Johann Bollen, Alberto Pepe and Huina Mao. Modelling Public Mood and Emotion: Twitter Sentiment and socio-economic phenomena. *In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[9] Luciano Barbosa and Junlan Feng. Robust Sentiment Detection on Twitter from Biased and Noisy Data. *In Proceedings of the international conference on Computational Linguistics (COLING)*, 2010.

[10] Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *In Proceedings of the Annual Meeting of the Association of Computational*

Linguistics (ACL), 2002.

[11] Rudy Prabowo and Mike Thelwall. Sentiment Analysis: A Combined Approach. Journal of Infometrics, Volume 3, Issue 2, April 2009, Pages 143-157, 2009.

[12] Samuel Brody and Nicholas Diakopoulus. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! Using Word Lengthening to Detect

[13] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. *In Proceedings of international conference on Language Resources and Evaluation (LREC)*, 2010.

[14] Theresa Wilson, Janyce Wiebe and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. *In the Annual Meeting of Association of Computational Linguistics: Human Language Technologies (ACL-HLT)*, 2005.

[15] Ian H. Witten, Eibe Frank & Mark A. Hall. Data Mining – Practical Machine Learning Tools and Techniques.

[19] Ricgard O. Duda, Peter E. Hart & David G. Stork: Pattern Classification.

[19] Steven Bird, Even Klein & Edward Loper. Natural Language Processing with Python.

[20] Soo-Min Kim and Eduard Hovy. Determining the Sentiment of Opinions. *In Proceedings of International Conference on Computational Linguistics (ICCL)*, 2004.