ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)

Available Online at www.ijetcr.org

Volume 8; Issue 4; July-August 2020; Page No. 01-10

TWITTER SENTIMENT ANALYSIS

Rakesh Ranjan (PhD)[1], Rahul Bind[2], Rishabh Sahu[3], Anurag[4]

[1,2,3,4] Department of Information Technology, Pranveer Singh Institute of Technology, Kanpur, (U.P.)

Conflicts of interest: Nil

Corresponding author: Rakesh Ranjan

Abstract

Abstract: Since an ever-increasing part of the population makes use of social media in their day-to-day lives, social media data is being analyzed in many different disciplines. The social media analytics process involves four distinct steps, data discovery, collection, preparation, and analysis. While there is a great deal of literature on the challenges and difficulties involving specific data analysis methods, there hardly exists research on the stages of data discovery, collection, and preparation. To address this gap, we conducted an extended and structured literature analysis through which we identified challenges addressed and solutions proposed. The literature search revealed that the volume of data was most often cited as a challenge by researchers. In contrast, other categories have received less attention. Based on the results of the literature search, we discuss the most important challenges for researchers and present potential solutions. The findings are used to extend an existing framework on social media analytics. The article provides benefits for researchers and practitioners who wish to collect and analyze social media data.[1]

Index Terms: Social media Scraping Behavior economics Sentiment analysis Opinion mining NLP Toolkits Software platforms

1 Introduction

This Data analysis is the process of applying organized and systematic statistical techniques to describe, recap, check and condense data. It is a multistep process that involves collecting, cleaning, organizing and analyzing. Data mining[1] is like applying techniques to mold data to suit our requirement. Data mining is needed because different sources like social media, transactions, public data, enterprises data etc. generates data of increasing volume, and it is important to handle and analyze such a big data. It won't be wrong to say that social media is something we live by. In the 21st century social media has been the game changer, be it advertising, politics or globalization, it has been estimated that data is increasing faster than before and by the year 2020; about 1.7 megabytes of additional data will be generated each instant for each person on the earth. More data has been generated in the past two years than ever before in the history of the mankind. It is clear from the fact that the numbers of internet users are now grown from millions to billions.

Database which is opted for the proposed study is from Twitter [2]. It is now day's very popular service which provides facility of microblogging. In this people write short messages generally less than 140 characters, about 11 words on average. It is appropriate for analysis as the number of messages is large. It is much easier task as compared to searching blogs from the net. The objective of the proposed analysis, 'Sentiment Analysis'[3], is the analysis of the enormous amount of data easily available from social media[4]. Algorithm generates an overall sentiment score from the inputted topic in terms of positive, negative or neutral, further it also works on finding the frequency of the words being used. Word cloud that is a pictorial representation of words based on frequency occurrence of words in the text is also generated.

Calculation is actualized utilizing R attributable to its component rich, thorough and expressive abilities for measurable information.

2.1 Purpose of this Project

This project of analyzing sentiments of tweets comes under the domain of "Pattern Classification" and "Data Mining"[5]. Both of these terms are very closely related and intertwined, and they can be formally defined as the process of discovering "useful" patterns in large set of data, either automatically (unsupervised)[6] or semi automatically (supervised). The project would heavily rely on techniques of "Natural Language Processing"[7] in extracting significant patterns and features from the large data set of tweets and on "Machine Learning" techniques[8] for accurately classifying individual unlabeled data samples (tweets) according to whichever pattern model best describes them.

2.2 Overview

This proposal is a web application which is used to analyze the tweets. We will be performing sentiment analysis in tweets and determine where it is positive, negative or neutral. This web application can be used by any organization office to review their works or by political leaders or by any others company to review about their products or brands. The main feature of our web application is that it helps to determine the opinion about the peoples on products, government work, politics or any other by analyzing the tweets. Our system is capable of training the new tweets taking reference to previously trained tweets. The computed or analyzed data[9] will be represented in various diagram such as Pie- chart, Bar graph and Word cloud.

2.3 Problem definition

The algorithm proposed works on Twitter Data, primarily it collects the tweets and then study it with the help of different statistical computing procedures[10]. Twitter account once registered and logged in, needs registering the application name on Twitter API to create our Copyright Form application which provide us the four legal credentials (API_key, API_secret, access token, access_token_secret)[11] required for connection establishment.

3 Brief History of Work Done

In this paper Sentiment analysis has been handled as a Natural Language Processing task at many levels of granularity. Starting from being a document level classification, it has been handled at the sentence level and more recently at the phrase level (Wilson et al., 2005; Agarwal et al., 2009). Microblog[12] data like Twitter, on which users post real time reactions to and opinions about "everything", poses newer and different challenges. Some of the early and recent results on sentiment analysis of Twitter data are by Go et al. (2009), (Birmingham and Smeaton, 2010) and Pak and Paroubek (2010).

Go et al. (2009) [13] use distant learning to acquire sentiment data. They use tweet sending in positive emotions like ":)" ":-)" as positive and negative emoticons like ":(" ":-(" as negative. They build models using Naive Bayes, Max Ent and Support Vector Machines (SVM), and they report SVM outperforms other classifiers. In terms of feature space, they try a Unigram, Bigram model in conjunction with parts-of-speech (POS) features. They note that the unigram model outperforms all other models. Specifically, bigrams and POS features do not help. Pak and Paroubek (2010)[14] collect data following a similar distant learning paradigm. They perform a different classification task though: subjective versus objective. For subjective data they collect the tweets ending with emoticons in the same manner as Go et al. (2009). For objective[15] data they crawl Twitter accounts of popular newspapers like "New York

Times", "Washington Posts" etc. They report that POS and bigrams both help (contrary to results presented by Go et al. (2009)). Both these approaches, however, are primarily based on ngram models. Moreover, the data they use for training and testing is collected by search queries and is therefore biased. In contrast, we present features that achieve a significant gain over a unigram baseline. In addition, we explore a different method of data representation and report significant improvement over the unigram models. Another contribution of this paper is that

we report results on manually annotated data that does not suffer from any known biases. Our data will be a random sample of streaming tweets unlike data collected by using specific queries. The size of our hand-labelled data will allow us to perform cross validation experiments and check forth variance in performance of the classifier across folds. Another significant effort for sentiment classification on Twitter data is by Barbosa and Feng (2010) [16]. Gamon (2004) [17] perform sentiment analysis on feedback data from Global Support Services survey.

For feature selection[18], Pang and Lee suggested to remove objective sentences by extracting subjective ones. They proposed a text-categorization technique that is able to identify subjective content using minimum cut. Gann et al. Selected 6799 tokens based on Twitter data, where each token is assigned a sentiment score, namely TSI (Total Sentiment Index), featuring itself as a positive token or a negative token. Specifically, a TSI for a certain token is computed as: $TSI = p - tp\ tn * n\ p + tp\ tn * n$ where p is the number of times a token appears in positive tweets and n is the number of times a token appears in negative tweets. Comparing to previous approaches in sentiment topics, additional findings by showed that adding the semantic feature produces better Recall (retrieved documents) to compute the score) in negative sentiment classification.

4 User Characteristics

Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text, speech, tweets and database sources through Natural Language Processing (NLP). Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral". It's also referred as subjectivity analysis, opinion mining, and appraisal extraction. The words opinion, sentiment, view and belief are used interchangeably but there are differences between them.

A conclusion open to dispute (because different experts have different opinions) View: subjective opinion Belief: deliberate acceptance and intellectual assent Sentiment: opinion representing one's feelings .Sentiment Analysis is a term that include many tasks such as sentiment extraction, sentiment classification, subjectivity classification, summarization of opinions or opinion spam detection[19] , among others. It aims to analyze people's sentiments, attitudes, opinions emotions, etc. towards elements such as, products, individuals, topics, organizations, and services.

5 System Analysis

5.1 Feasibility Study:

A feasibility study[20] is a preliminary study which investigates the information of prospective users and determines the resources requirements, costs, benefits and feasibility of proposed system. The main objectives of the feasibility study are to determine whether the project would be feasible in terms of economic feasibility, technical feasibility[21] and operational feasibility[22] and schedule feasibility or not. It is to make sure that the input data which are required for the project are available. Thus, we evaluated the feasibility of the system in terms of the following categories:

- Technical feasibility
- Operational feasibility
- Economic feasibility
- Schedule feasibility

5.1.1 Technical Feasibility

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at the point in time there is no any detailed designed of the system, making it difficult to access issues like performance, costs. Is Our system is technically feasible since all the required tools are easily available. Python[23] makes the system more user and developer friendly and although all tools seem to be easily available there are challenges too.

Opinion:

5.1.2 Operational Feasibility

Proposed project is beneficial only if it can be turned into information systems that will meet the operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? The proposed was to make a simplified web application. It is simpler to operate and can be used in any webpages. It is free and not costly to operate.

5.1.3 Economic Feasibility

Economic feasibility[24]attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. These could increase improvement in product quality, better decision making, and timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information. This is a GUI-based application. Creation of application is not costly.

5.1.4 Schedule Feasibility

Schedule feasibility[25]is a measure how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some project is initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable. A minor deviation can be encountered in the original schedule decided at the beginning of the project. The application development is feasible in terms of schedule.

5.2 Requirement Definition:

After the extensive analysis of the problems in the system, we are familiarized with the requirement that the current system needs. The requirement that the system needs is categorized into the functional requirement[26] and non-functional requirements[27]. These requirements are listed below:

5.2.1 Functional Requirements

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows: -System should be able to process new tweets stored in database after retrieval -System should be able to analyze data and classify each tweet polarity

5.2.2Non-Functional Requirements

Non-functional requirements is a description of features, characteristics and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. The non-functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. Based on these the non-functional requirements are as follows:

- User friendly
- System should provide better accuracy
- To perform with efficient throughput and response time

5.3 Study of Current System

There are primarily two types of approaches for sentiment classification of opinionated texts: Using a Machine learning based text classifier such as Naive Bayes Using Natural Language Processing[28], we will be using those machine learning and natural language processing for sentiment analysis of tweets.

5.3.1 Machine Learning

Introduction

The machine learning based text classifiers are a kind of supervised machine learning paradigm, where the classifier needs to be trained on some labelled training data before it can be applied to actual classification task. The training data is usually an extracted portion of the original data hand labelled manually. After suitable training they can be used on the actual test data. The Naive Bayes is a statistical classifier whereas Support Vector Machine is a kind of vector space classifier. The statistical text classifier scheme of Naive Bayes (NB) can be adapted to be used for sentiment classification problem as it can be visualized as a 2-class text classification problem: in positive and negative classes. Support Vector

machine (SVM)[29] is a kind of vector space model-based classifier which requires that the text documents should be transformed to feature vectors before they are used for classification. Usually the text documents are transformed to multidimensional vectors. The entire problem of classification is then classifying every text document represented as a vector into a particular class. It is a type of large margin classifier. Here the goal is to find a decision boundary between two classes that is maximally far from any document in the training data. This approach needs good classifier such as Naive Bayes. A training set for each class There are various training sets available on Internet such as Movie Reviews data set, twitter dataset, etc. Class can be Positive, negative. For both the classes we need training data sets.

Machine Learning is undeniably one of the most influential and powerful technologies in today's world. More importantly, we are far from seeing its full potential. There's no doubt, it will continue to be making headlines for the foreseeable future. This article is designed as an introduction to the Machine Learning concepts, covering all the fundamental ideas without being too high level. Machine learning is a tool for turning information into knowledge. In the past 50 years, there has been an explosion of data. This mass of data is useless unless we analyse it and find the patterns hidden within. Machine learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover. The hidden patterns and knowledge about a problem can be used to predict future events and perform all kinds of complex decision making.

We are drowning in information and starving for knowledge — John Naisbitt

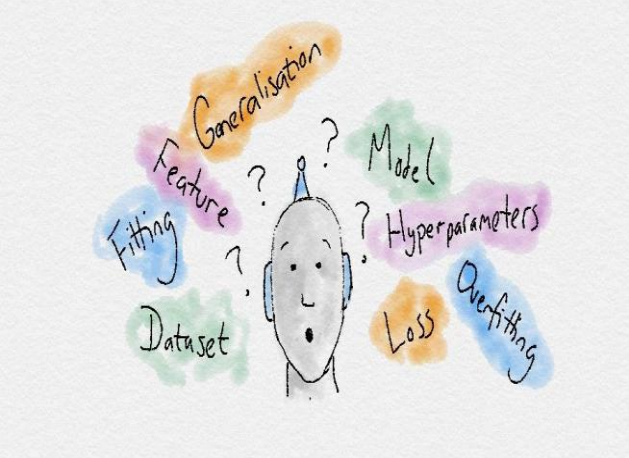
Most of us are unaware that we already interact with Machine Learning every single day. Every time we Google something, listen to a song or even take a photo, Machine Learning is becoming part of the engine behind it, constantly learning and improving from every interaction. It's also behind world-changing advances like detecting cancer, creating new drugs and self-driving cars.

The reason that Machine Learning is so exciting, is because it is a step away from all our previous rule-based systems of:

if(x = y): do z

Traditionally, software engineering combined human created *rules* with *data* to **create answers to a problem**. Instead, machine learning uses *data* and *answers* to **discover the rules behind a problem**. (Chollet, 2017). To learn the rules governing a phenomenon, machines have to go through a **learning process,** trying different rules and learning from how well they perform. Hence, why it's known as Machine Learning. There are multiple forms of Machine Learning; supervised, unsupervised, semi-supervised and reinforcement learning. Each form of Machine Learning has differing approaches, but they all follow the same underlying process and theory. This explanation covers the general Machine Leaning concept and then focusses in on each approach.

Terminology



- **Dataset**: A set of data examples that contain features important to solving the problem.
- **Features**: Important pieces of data that help us understand a problem. These are fed in to a Machine Learning algorithm to help it learn.
- **Model**: The representation (internal model) of a phenomenon that a Machine Learning algorithm has learnt. It learns this from the data it is shown during training. The model is the output you get after training an algorithm. For example, a decision tree algorithm would be trained and produce a decision tree model.

Process

1. **Data Collection:** Collect the data that the algorithm will learn from.

2. **Data Preparation:** Format and engineer the data into the optimal format, extracting important features and performing dimensionality reduction.

3. **Training:** Also known as the fitting stage, this is where the Machine Learning algorithm actually learns by showing it the data that has been collected and prepared.

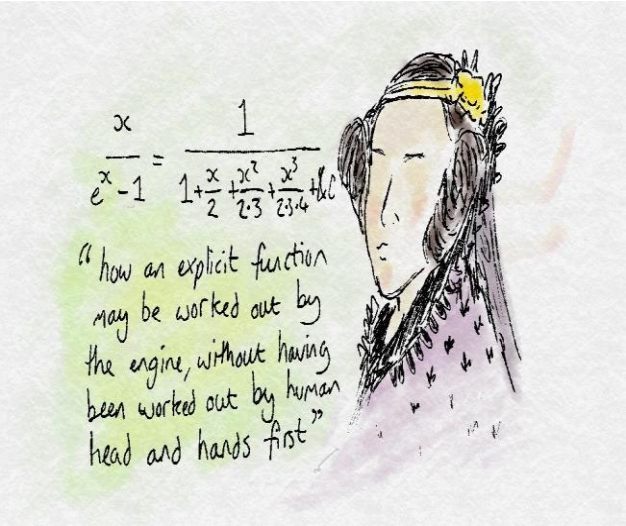
4. **Evaluation:** Test the model to see how well it performs.

5. **Tuning:** Fine tune the model to maximise it's performance.

Background Theory

Origins

The Analytical Engine weaves algebraic patterns just as the Jacquard weaves flowers and leaves — Ada Lovelace



Ada Lovelace[30], one of the founders of computing, and perhaps the first computer programmer, realised that **anything in the world could be described with math**. More importantly, this meant a mathematical formula can be created to derive the relationship representing any phenomenon. Ada Lovelace realised that **machines had the potential to understand the world without the need for human assistance**. Around 200 years later, these fundamental ideas are critical in Machine Learning. No matter what the problem is, it's information can be plotted onto a graph as data points. Machine Learning then tries to find the mathematical patterns and relationships hidden within the original information.

Probability Theory

Probability is orderly opinion… inference from data is nothing other than the revision of such opinion in the light of relevant new information — Thomas Bayes

Another mathematician, Thomas Bayes, founded ideas that are essential in the probability theory that is manifested into Machine Learning. We live in a probabilistic world. Everything that happens has uncertainty attached to it. The Bayesian interpretation of probability is what Machine Learning is based upon. Bayesian probability means that we think of probability as **quantifying the uncertainty** of an event. Because of this, we have to base our probabilities on **the information available about an event**, rather than counting the number of repeated trials. For example, when predicting a football match, instead of counting the total amount of times Manchester United have won against Liverpool, a Bayesian approach would use **relevant information** such as the current form, league placing and starting team. The benefit of taking this approach is that probabilities can still be assigned to rare events, as the decision-making process is based on **relevant features** and **reasoning**.

Machine Learning Approaches

There are many approaches that can be taken when conducting Machine Learning. They are usually grouped into the areas listed below. Supervised and Unsupervised are well established approaches and the most commonly used. Semi-supervised and Reinforcement Learning are newer and more complex but have shown impressive results.

The **No Free Lunch theorem** is famous in Machine Learning. It states that there is no single algorithm that will work well for all tasks. Each task that you try to solve has its own idiosyncrasies. Therefore, there are lots of algorithms and approaches to suit each problem individual quirks. Plenty more styles of Machine

Learning and AI will keep being introduced that best fit different problems.

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

5.3.2 Python (programming language)

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and objectoriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural,) object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward compatible, and much Python 2 code does not run unmodified on Python 3. The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported. Python interpreters are available for many operating systems. A global community of programmers develops and maintains Python, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and C Python development

API documentation generators

Python API documentation generators include:

- Epydoc
- HeaderDoc
- Sphinx
- pydoc

Libraries

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of November 2019, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 200,000 packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Test frameworks
- Networking
- Automation
- Web scraping
- System administration
- Documentation
- Scientific computing
- Text processing
- Image processing
- Machine learning
- Data analytics

5.3.3 Decision Tree

Decision Tree: Decision tree[31] is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Decision Tree Representation:

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node.

Construction of Decision Tree:

A tree can be *"learned"* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called *recursive partitioning*. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification. The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf.(in this case Yes or No).

For example, the instance

(Outlook = Rain, Temperature = Hot, Humidity = High, Wind = Strong)

would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance. In other words, we can say that decision tree represent a disjunction of conjunctions of constraints on the attribute values of instances.

(Outlook = Sunny ^ Humidity = Normal) v (Outlook = Overcast) v (Outlook = Rain ^ Wind = Weak)

5.3.4 Naïve Bayes Classifier (NB): -

The Naïve Bayes classifier[32] is the simplest and most commonly used classifier. Naïve Bayes classification model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the BOWs feature extraction which ignores the position of the word in the document. It uses Bayes Theorem [34] to predict the probability that a given feature set belongs to a particular label.

$$P(\text{label}|\text{features}) = P(\text{label}) * P(\text{features}|\text{label}) \; P(\text{features})$$

where, P(label): - is the prior probability of a label or the likelihood that a random feature set the label. P(features|label): - is the prior probability that a given feature set is being classified as a label. P(features): - is the prior probability that a given feature set is occurred. Given the Naïve assumption which states that all features are independent, the equation could be rewritten as follows:

$$P(\text{label}|\text{features}) = P(\text{label}) * P(f1|\text{label}) * \ldots\ldots\ldots * P(fn|\text{label}) \; P(\text{features})$$

Multinomial Naïve Bayes Classifier

Accuracy – around 75%

Algorithm:

I. Dictionary generation

Count occurrence of all word in our whole data set and make a dictionary of some most frequent words.

II. Feature set generation

All document is represented as a feature vector over the space of dictionary words. For each document, keep track of dictionary words along with their number of occurrences in that document.

5.3.5 Natural Language Processing

Natural language processing (**NLP**) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.

Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural language generation.

Rule-based vs. statistical NLP

In the early days, many language-processing systems were designed by hand-coding a set of rules: such as by writing grammars or devising heuristic rules for stemming.

Since the so-called "statistical revolution"**[33]** in the late 1980s and mid-1990s, much natural language processing research has relied heavily on machine learning. The machine-learning paradigm calls instead for using statistical inference to automatically learn such rules through the analysis of large *corpora* (the plural form of *corpus*, is a set of documents, possibly with human or computer annotations) of typical real-world examples.

5.4 CHALLENGES IN SENTIMENT ANALYSIS:

1. Identifying subjective parts of text
2. Domain dependence
3. Sarcasm Detection
4. Thwarted expressions
5. Explicit Negation of sentiment
6. Order dependence
7. Entity Recognition
8. Applying sentiment analysis to Facebook messages
9. Internationalization
10. Handling comparisons

Conclusion

The experimental studies performed through the chapters, successfully show that hybridizing the existing machine learning analysis and lexical analysis techniques for sentiment classification yield comparatively outperforming accurate results. For all the datasets used, we recorded consistent accuracy of almost 90%. The first method that we approached for our problem is Naïve Bayes. It is mainly based on the independence assumption. Training is very easy and fast in this approach each attribute in each class is considered separately. Testing is straightforward, calculating the conditional probabilities from the data available. One of the major tasks is to find the sentiment polarities which is very important in this approach to obtain desired output. In this Naïve Bayes approach, we only considered the words that are available in our dataset and calculated their conditional probabilities. We have obtained successful results after applying this approach to our problem. Clearly from the success of Hybrid Naive Bayes, it can positively be applied over other related sentiment analysis applications like financial sentiment analysis (stock market, opinion mining), customer feedback services, and etc.

ACKNOWLEDGMENT

It gives us a great sense of pleasure to present the report of B. Tech Project undertaken during B. Tech Final Year. We owe special debt gratitude to **Mr. Rakesh Ranjan** (Professor) Department of Information Technology, Pranveer Singh institute of Technology Kanpur for his constant support and guidance throughout the course of our work. Her sincerity thoroughness and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavors have been seen light of the day.

We also take the opportunity to acknowledge the contribution of **Mr. Piyush Bhushan Singh** Head of Department (Information Technology) for his full support and assistance during the development of the project.

References

1. https://www.sas.com/en_in/insights/analytics/data-mining.html
2. https://en.wikipedia.org/wiki/twitter
3. https://www.lexalytics.com/technology/sentiment-analysis
4. https://www.thebalancesmb.com/what-is-social-media-2890301

5. https://www.microstrategy.com/us/resources/introductory-guides/data-mining- explained
6. https://www.geeksforgeeks.org/supervised-unsupervised-learning/
7. https://machinelearningmastery.com/natural-language-processing/
8. https://www.simplilearn.com/techniques-of-machine-learningtutorial?clickid=1kfzs-yxmxyotuxwux0mo3q3uki0hu0ldxlyue0&irgwc=1&utm_medium=123201&utm_campaign=online%20tracking%20link&utm_source=ir
9. https://www.predictiveanalyticstoday.com/data-analysis/
10. https://www.sciencedirect.com/science/article/pii/s0022030290786997
11. https://auth0.com/docs/tokens/concepts/access-tokens
12. https://sproutsocial.com/glossary/microblog/
13. https://www.researchgate.net/publication/228523135_Twitter_sentiment_classification_using_distant_supervision
14. https://www.researchgate.net/publication/220746311_Twitter_as_a_Corpus_for_Sentiment_Analysis_and_Opinion_Mining
15. https://testprepnerds.com/nclex/subjective-vs-objectivedata/#:~:text=Objective%20Data%20is%20the%20physical,%2C%20tests%2C%20an d%20physical%20exams.
16. https://www.researchgate.net/publication/221101683_Robust_Sentiment_Detection_on_Twitter_from_Biased_and_Noisy_Data
17. https://www.researchgate.net/publication/215470705_Sentiment_classification_on_customer_feedback_data_Noisy_data_large_feature_vectors_and_the_role_of_linguistic_analysis
18. https://towardsdatascience.com/feature-selection-techniques-in-machinelearning-with-python-
19. f24e7da3f36e#:~:text=Feature%20Selection%20is%20the%20process,learn%20based%20on%20irrelevant%20features.
20. https://www.mailchannels.com/what-is-spamfiltering/#:~:text=What%20is%20Spam% 20 Filtering %3F,they%20aren't%20distributi ng %20spam.
21. https://www.geeksforgeeks.org/differences-between-verification-andvalidation/#:~: text=Verification%20means%20Are%20we%20building,developing%20is%20t he%20right% 20product.
22. https://www.sciencedirect.com/topics/computer-science/technical-feasibility
23. https://www.wikiwand.com/en/Feasibility_study#:~:text=Operational%20feasibility%20study,analysis%20phase%20of%20system%20development.
24. https://www.wikiwand.com/en/Python_(programming_language)
25. http://www.businessdictionary.com/definition/economic-feasibility.html
26. https://www.slideshare.net/UdaraSeneviratne/scedule-feasibility
27. https://www.guru99.com/functional-requirement-specificationexample.html#:~:text=A% 20 Functional%20Requirement%20(FR)%20is,%2C%20its%20behavi or%2C%20and%20outputs.
28. https://www.scaledagileframework.com/nonfunctionalrequirements/#:~:text=Nonfunctional%20Requirements%20(NFRs)%20define%20system,system%20across%20the%20different%20backlogs.
29. https://becominghuman.ai/a-simple-introduction-to-natural-language-processingea66a1747b32
30. https://www.wikiwand.com/en/Support_vector_machine
31. https://www.wikiwand.com/en/Ada_Lovelace
32. http://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decisiondc506a403aeb#:~:text=A%20decision%20tree%20is%20a%20flowchartlike%20structure%20in%20which,taken%20after%20computing%20all%20attributes).
33. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
34. https://medium.com/friendly-data/machine-learning-vs-rule-based-systems-in-nlp5476de53c3b8