IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2021
Impact Factor: 5.731

Fast and Accurate Sentiment Classification Using NLTK and Naïve Bayes Model

Mr. Himanshu Sekhar Acharya
Assistant Professor, Department of Computer Science
Ravenshaw University, Cuttack, Odisha

Abstract: *In today's world, Social Networking website like Twitter, Facebook, Linkedin, etc. plays a very significant role. Twitter is a micro-blogging platform which provides a tremendous amount of data which can be used for various application of sentiment Analysis like predictions, review, elections, marketing, etc. Sentiment Analysis is a process of extracting information from large amount of data, and classifies them into different classes called sentiments. Python is simple yet powerful, high-level, interpreted and dynamic programming language, which is well known for its functionality of processing natural language data by using NLTK (Natural Language Toolkit). NLTK is a library of python, which provides a base for building programs and classification of data. NLTK also provide graphical demonstration for representing various results or trends and it also provide sample data to train and test various classifier respectively. Sentiment classification aims to automatically predict sentiment polarity of users publishing sentiment data. Although traditional classification algorithm can be used to train sentiment classifiers from manually labelled text data, the labelling work can be time-consuming and expensive. Meanwhile, users often use some different words when they express sentiment in different domains. If we directly apply a classifier trained in one domain to other domains, the performance will be very low due to the difference between these domains. In this work, we develop a general solution to sentiment classification when we do not have any labels in target domain but have some labelled data in a different domain, regarded as source domain.*

Keywords: Natural Language Toolkit

I. INTRODUCTION

A large amount of data that is generated today is unstructured, which requires processing to generate insights. Some examples of unstructured data are news articles, posts on social media, and search history. The process of analyzing natural language and making sense out of it falls under the field of Natural Language Processing (NLP). Sentiment analysis is a common NLP task, which involves classifying texts or parts of texts into a pre-defined sentiment. I have used the Natural Language Toolkit(NLTK), a commonly used NLP library in Python, to analyze textual data.

Sentiment analysis involves extraction of subjective information from documents like online reviews to determine the polarity with respect to certain objects. It is useful for identifying trends of public opinion in the social media, for the purpose of marketing and consumer research. It has its uses in getting customer feedback about new product launches, political campaigns and even in financial markets. It aims to determine the attitude of a speaker or a writer with respect to some topic or simply the contextual polarity of a document. Sentiment classification has number of application which is helpful in business, marketing and increasing sell of the product.

The goal of Sentiment Analysis is to harness this data in order to obtain important information regarding public opinion, that would help make smarter business decisions, political campaigns and better product consumption. In this project we present a supervised sentiment classification model based on the Naive Bayes algorithm. Naive Bayes is a very simple probabilistic model that tends to work well on text classifications and usually takes orders of magnitude less time to train when compared to models like support vector machines. We will show in this project that a high degree of accuracy can be obtained using Naive Bayes classifier, which is comparable to the current state of the art models in sentiment classification.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2021
Impact Factor: 5.731

In this project, we prepared a dataset of sample tweets from the NLTK package for NLP with different data cleaning methods. Once the dataset is ready for processing, we trained a model on pre-classified tweets and use the model to classify the sample tweets as test data into negative and positives sentiments.

II. Natural Language Processing

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. The goal of natural language processing is to allow that kind of interaction so that non-programmers can obtain useful information from computing systems. Natural language processing also includes the ability to draw insights from data contained in emails, videos, and other unstructured material. The various aspects of NLP include Parsing, Machine Translation, Language Modeling, Machine Learning, Semantic Analysis etc. In this project we only focus on semantic analysis aspect of NLP using NLTK.

Natural language processing refers to the use and capability of systems to process sentences in a natural language such as English, rather than in a specialized artificial computer language such as Java. Broadly construed, Natural language processing (with respect to the interpretation side) is considered to involve the following subtopics:

1. Syntactic analysis
2. Semantic analysis
3. Pragmatics

Syntactic analysis includes consideration of morphological and syntactic knowledge on the part of the natural language processor, semantic analysis includes consideration of semantic knowledge, and pragmatics includes consideration of pragmatic, discourse, and world knowledge. We perform syntactic analysis and semantic analysis by benefitting from the structures in the WordNet library and comparing them using NLTK. But, Pragmatics still remains out of the domain of this approach.

III. Natural Language Toolkit

NLTK is an open source natural language processing (NLP) platform available for Python. NLTK is a community driven project and is available for use on Linux, Mac OS X and Windows.

NLTK comes with an inbuilt sentiment analyser module that can analyse a piece of text and classify the sentences under positive, negative and neutral polarity of sentiments.

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. NLTK includes graphical demonstrations and sample data.

NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. We discuss how we can perform semantic analysis in NLP using NLTK as a platform for different corpora. Adequate representation of natural language semantics requires access to vast amounts of common sense and domain-specific world knowledge. We focus our efforts on using WordNet as a preferred corpora for using NLTK.

IV. WordNet

Because meaningful sentences are composed of meaningful words, any system that hopes to process natural languages as people do must have information about words and their meanings. This information is traditionally provided through dictionaries, and machine-readable dictionaries are now widely available. But dictionary entries evolved for the convenience of human readers, not for machines. WordNet provides a more effective combination of traditional lexicographic information and modern computing. WordNet is an online lexical database designed for use under program control. English nouns, verbs, adjectives, and adverbs are organized into sets of synonyms, each

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2021
Impact Factor: 5.731

representing a lexicalized concept. Semantic relations link the synonym sets. Using NLTK and WordNet, we can form semantic relations and perform semantic analysis on texts, strings and documents.

WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing. WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms of strings of letters but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than meaning similarity.

V. Naive Bayes Classifier

A Naive Bayes classifier is a simple probabilistic model based on the Bayes rule along with a strong independence assumption. The Naive Bayes model involves a simplifying conditional independence assumption. That is given a class (positive or negative), the words are conditionally independent of each other. This assumption does not affect the accuracy in text classification by much but makes really fast classification algorithms applicable for the problem. In this case, the maximum likelihood probability of a word belonging to a particular class is given by the expression:

$P(x_i|c)$ = Count of x_i in documents of class c / Total no of words in documents of class c

The frequency counts of the words are stored in hash tables during the training phase.

According to the Bayes Rule, the probability of a particular document belonging to a class c_i is given by,

$P(c_i|d) = \{ P(d|c_i) * P(c_i) \} / P(d)$

If we use the simplifying conditional independence assumption, that given a class (positive or negative), the words are conditionally independent of each other. Due to this simplifying assumption the model is termed as "naive".

$P(c_i|d) = \{(\Pi P(x_i|c_j)) * P(c_j)\} / P(d)$

Here the x_is are the individual words of the document. The classifier outputs the class with the maximum posterior probability. We also remove duplicate words from the document, they don't add any additional information; this type of naive bayes algorithm is called Bernoulli Naive Bayes. Including just the presence of a word instead of the count has been found to improve performance marginally, when there is a large number of training examples.

VI. Data : NLTK Corpora

A publicly available dataset of tweets from the Natural Language Toolkit Corpus Library is used. It has a set of 30,000 highly polar tweets for training, and 10,000 for testing. Both the training and test sets have an equal number of positive and negative tweets. We choose tweets as our data set because it covers a wide range of human emotions and captures most of the adjectives relevant to sentiment classification. Also, most existing research on sentiment classification uses tweet data for benchmarking. We used the 30,000 documents in the training set to build our supervised learning model. The other 10,000 were used for evaluating the accuracy of our classifier.

VII. Data Preprocessing

Data obtained from twitter is not fit for extracting features. Mostly tweets consists of message along with usernames, empty space, special character, stop words, emoticons, abbreviations, hash tags, time stamps, URL's, etc. Thus to make this data fit for mining we pre-process the dataset by using various function of NLTK.IN pre-processing. First task is to extract messages from the tweet, then remove all empty spaces, stop words (like is, a, the, he, them, etc), hash tags, repeating words, URL's etc. Then all emoticons and abbreviations are replaced with their corresponding meanings with happy or laugh. Once we are done with it, we are ready with processed tweet which is provided to classifier for required results. Sample tweet and processed tweet. Cleaning of twitter data is necessary, since tweets contain several syntactic features that may not be useful for analysis. The pre-processing is done in such a way that data represented only in terms of words that can easily classify the class.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2021
Impact Factor: 5.731

Figure 1: Steps involved in Sentiment Analysis

We create a code in python in which we define a function which will be used to obtain processed tweet. This data is used to train the classifier which we are going to build. To collect this data we use NLTK library of python. NLTK consist of corpora which is very large and consists of structured set of text files which are used to perform analysis In these corpora there are various types of text files like quotes, reviews, chat, history, tweets etc. From these corpora selected files of tweet samples are processed for our training purpose.

7.1 Tokenization

Language in its original form cannot be accurately processed by a machine, so you need to process the language to make it easier for the machine to understand. The first part of making sense of the data is through a process called tokenization, or splitting strings into smaller parts called tokens.

A token is a sequence of characters in text that serves as a unit. Based on how you create the tokens, they may consist of words, emoticons, hashtags, links, or even individual characters. A basic way of breaking language into tokens is by splitting the text based on whitespace and punctuation.

Figure 2: Tokenization

7.2 Normalization

Words have different forms for instance, "ran", "runs", and "running" are various forms of the same verb, "run". Depending on the requirement of the analysis, all of these versions may need to be converted to the same form, "run". Normalization in NLP is the process of converting a word to its canonical form.

Normalization helps group together words with the same meaning but different forms. Without normalization, "ran", "runs", and "running" would be treated as different words, even though we may want them to be treated as the same word. In this process, It is explored about stemming and lemmatization, which are two popular techniques of normalization.

Stemming is a process of removing affixes from a word. Stemming, working with only simple verb forms, is a heuristic process that removes the ends of words.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2021
Impact Factor: 5.731

Lemmatization is a process in which a word normalizes with the context of vocabulary and morphological analysis of words in text. The lemmatization algorithm analyzes the structure of the word and its context to convert it to a normalized form. Therefore, it comes at a cost of speed. A comparison of stemming and lemmatization ultimately comes down to a tradeoff between speed and accuracy. Wordnet is a lexical database available in NLTK for the English language that helps the script determine the base word.

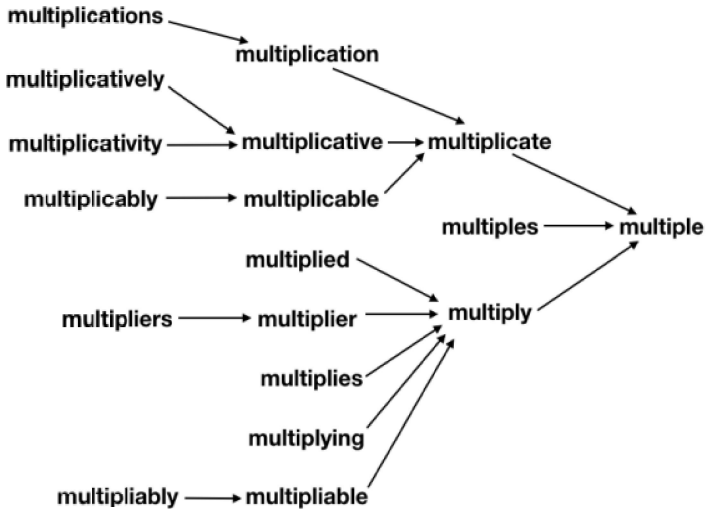

Figure 3: Lemmatization process

7.3 Removal of Noise and Stop Words

Noise is any part of the text that does not add meaning or information to data. Noise is specific to each project, so what constitutes noise in one project may not be in a different project. For instance, the most common words in a language are called*stop words*. Some examples of stop words are "is", "the", and "a". They are generally irrelevant when processing language, unless a specific use case warrants their inclusion.

- **Hyperlinks** - All hyperlinks in Twitter are converted to the URL shortener t.co. Therefore, keeping them in the text processing would not add any value to the analysis.
- **Twitter handles in replies** - These Twitter usernames are preceded by a @ symbol, which does not convey any meaning.
- **Punctuation and special characters** - While these often provide context to textual data, this context is often difficult to process. For simplicity, you will remove all punctuation and special characters from tweets.

To remove first search for a substring that matches a URL starting with http:// or https://, followed by letters, numbers or special characters. Once a pattern is matched, the .sub() method replaces it with an empty string.hyperlinks. Similarly, to remove @ mentions, the code substitutes the relevant part of text using regular expressions. The code uses the re library to search @ symbols, followed by numbers, letters, or _, and replaces them with an empty string.

Finally punctuation are removed using the library string. In addition to this, It also removed stop words using a built-in set of stop words in NLTK, which needs to be downloaded separately.

7.4 Determining Word Density

The most basic form of analysis on textual data is to take out the word frequency. A single tweet is too small of an entity to find out the distribution of words, hence, the analysis of the frequency of words would be done on all positive tweets. In code It has defined a generator function, named get_all_words, that takes a list of tweets as an argument to provide a list of words in all of the tweet tokens joined.

After compiling all words in the sample of tweets, the most common words can be found out using the FreqDist class of NLTK. The most_common() method lists the words which occur most frequently in the data.From this data, emoticon entities form some of the most common parts of positive tweets.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2021
Impact Factor: 5.731

To summarize, the tweets are extracted from NLTK, tokenized, normalized, and cleaned up the tweets for using in the model. Finally, It can also be looked at the frequencies of tokens in the data and checked the frequencies of the top ten tokens.

VIII. Model Preparation

Sentiment analysis is a process of identifying an attitude of the author on a topic that is being written about. It have been created a training data set to train a model. It is a supervised learning machine learning process, which requires to associate each dataset with a "sentiment" for training. In this project, the model used the "positive" and "negative" sentiments.

Sentiment analysis can be used to categorize text into a variety of sentiments. For simplicity and availability of the training dataset, this project helps to train my model in only two categories, positive and negative.

A model is a description of a system using rules and equations. It may be as simple as an equation which predicts the weight of a person, given their height. A sentiment analysis model that you will build would associate tweets with a positive or a negative sentiment. We need to split my dataset into two parts. The purpose of the first part is to build the model, whereas the next part tests the performance of the model.

First, the data have prepared to be fed into the model. Then the Naïve Bayes Classifier in NLTK is called to perform the modeling exercise. Notice that the model requires not just a list of words in a tweet, but a Python dictionary with words as keys and True as values. The following function makes a generator function to change the format of the cleaned data.

This code attaches a Positive or Negative label to each tweet. It then creates a dataset by joining the positive and negative tweets. By default, the data contains all positive tweets followed by all negative tweets in sequence. When training the model, It should provide a sample of data that does not contain any bias. To avoid bias, to randomly arrange the data using the .shuffle() method of random module is added to the code.

Finally, the code splits the shuffled data into a ratio of 70:30 for training and testing, respectively. Since the number of tweets is 30000, the first 20000 tweets from the shuffled dataset is used for training the model and the final 10000 for testing the model.

IX. Building and Testing The Model

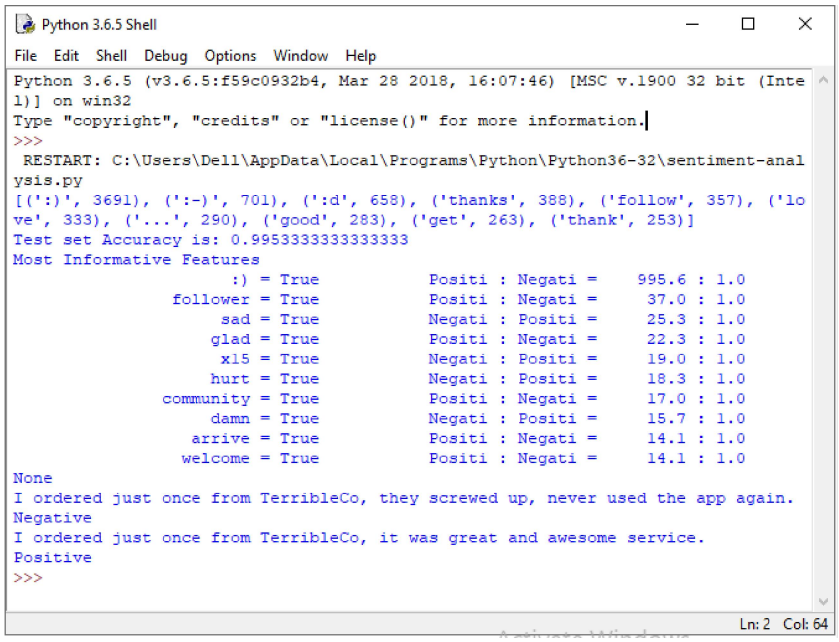

Figure 4: Output of the Model

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2021
Impact Factor: 5.731

Finally, Naive Bayes Classifier class implemented to build the model. The .train() method of Naive Bayes Classifier is used to train the model and the .accuracy() method to test the model on the testing data and determine the accuracy of the model.

Accuracy is defined as the percentage of tweets in the testing dataset for which the model was correctly able to predict the sentiment. After testing about 99.5% accuracy has been on the test set which is much higher than Support Vector Classifier, Gaussian Naive Bayes Classifier, Multinomial Naive Bayes Classifier.

X. System Workflow

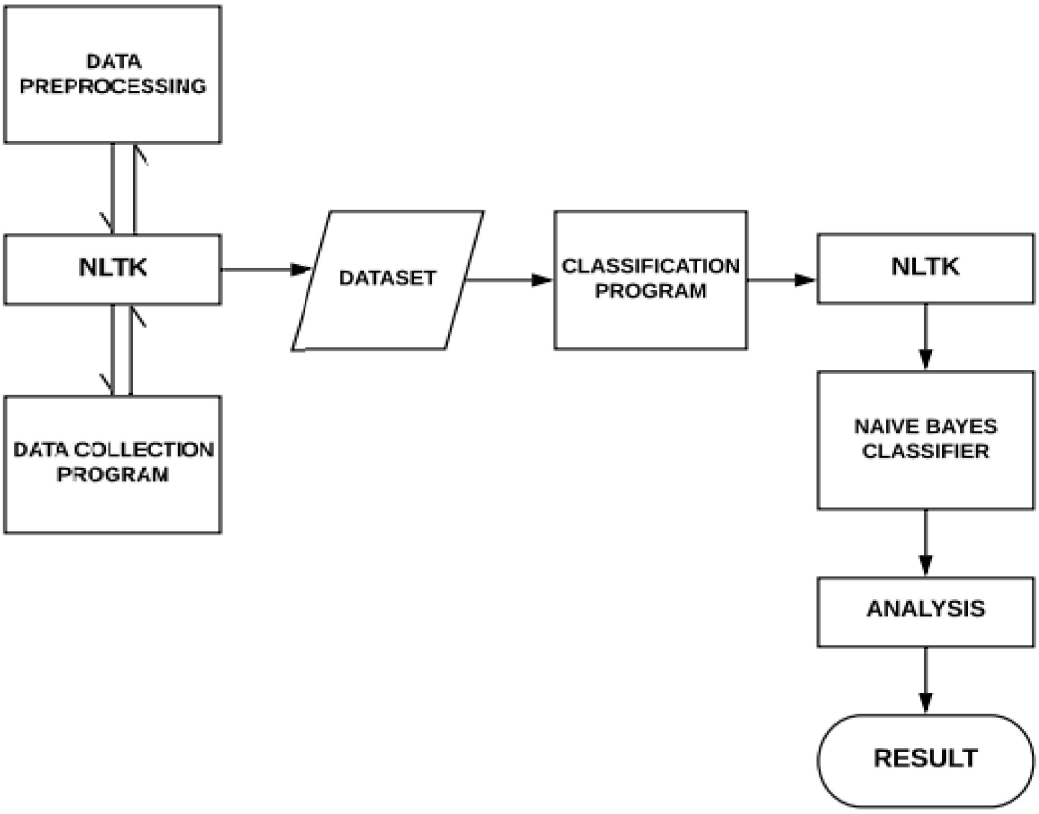

Figure 5: System workflow

XI. Future Work

In this project, The system computes the frequency of each term in tweet. A machine learning supervised approach implemented to obtain the results. Twitter is large source of data, which make it more attractive for performing sentiment analysis. We perform analysis on dataset which is publicly available by Natural Language Toolkit corpora library consist of about 30,000 tweets, so that we analyze the results, understand the patterns and give a review on people opinion. It can be concluded that more the cleaner data, more accurate results can be obtained. A web based application can be made for work in future. Existing system that can deal with sentence of multiple meanings can be improved and can also increase the classification categories so that we can get better results. The system can also improved and supervised to work on multi languages like Hindi, Spanish and Arabic to provide sentiment analysis to more local.

XII. Conclusion

Our results show that a simple Naive Bayes classifier can be enhanced to match the classification accuracy of more complicated models for sentiment analysis by choosing the right type of features and removing noise by appropriate

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2021
Impact Factor: 5.731

feature selection. Naive Bayes classifiers due to their conditional independence assumptions are extremely fast to train and can scale over large datasets. They are also robust to noise and less prone to overfitting. Ease of implementation is also a major advantage of Naive Bayes and NLTK libraries. They were thought to be less accurate than their more sophisticated counterparts like support vector machines and logistic regression but it is shown through this project that a significantly high accuracy can be achieved. The ideas used in this project can also be applied to the more general domain of text classification.

REFERENCES

[1]. https://www.researchgate.net/publication/220482883_NLTK_the_Natural_Language_Toolkit
[2]. http://ijcsit.com/docs/Volume%206/vol6issue06/ijcsit20150606134.pdf
[3]. https://www.kaggle.com/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews
[4]. https://www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-python
[5]. https://pdfs.semanticscholar.org/c151/dfad8c1bf88b0afc716758c77d533ded7dd0.pdf
[6]. https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk
[7]. Basic Sentiment Analysis using NLTK - Towards Data Science
[8]. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/