**Sentence Level Text Classification Using Naïve Bayes And SVM**

PROJECT REPORT

OF

**19AIE205**

*Submitted by* ***Team-06***

**BL.EN.U4AIE20031- Kunisetty Jaswanth**

**BL.EN.U4AIE20032- Masabattula Teja Nikhil**

**BL.EN.U4AIE20036- Mutyala Sai Sri Siddhartha**

***In partial fulfillment for the award of the degree***

***Of***

**BACHELOR OF TECHNOLOGY**

IN

ARTIFICIAL INTELLIGENCE ENGINEERING

**A picture containing text, clipart

Description automatically generated**

AMRITA SCHOOL OF ENGINEERING, BANGALORE  
AMRITA VISHWA VIDYAPEETHAM  
BANGALORE

JANUARY 2022

1. **ABSTRACT**

Now a days natural language processing is becoming more important and showing its influence in different fields which include robotic automation, bio informatics, e-commerce reviews understanding. Sentimental analysis is one of the problem statement in natural language processing which is capable of understanding the feelings or opinions that underlie a text. This project is on the sentence level text classification model which can identify whether the given sentence represents a positive sentiment or a negative sentiment. Consider the reviews on a particular product or a movie, the reviews can be of two types: structured reviews which are predefined, unstructured reviews which are written reviews. Structured reviews are easier to analyse or automate to understand, whereas the textual reviews require the understanding of each and every word, here comes problem of the computation. The classification of the reviews can be done in different ways which include Deep learning, machine learning. In this project a machine learning (Naïve Bayes, SVM) based movie review classification model has been analysed and experimental results are been compared.

**Key words**: Natural Language Processing (NLP), Stemming, Count Vectorizer, Tokens, Hard expressions, corpus, stop words

1. **TABLE OF CONTENTS**

**Page.no**

1. **Abstract - 2**
2. **Contents - 3**
3. **List of figures - 4**
4. **Libraries Used - 5**
5. **Introduction - 6**
6. **Concepts Used - 7**
7. **Algorithm - 9**
8. **Results - 11**
9. **Conclusion - 11**
10. **References - 12**
11. **LIST OF FIGURES**

**Figure Description Page.no**

**1 Stemming of a sentence 7**

**2 Token Generation from sentences 8**

**3 Bag-of-words Representation 8**

**4 Stanford Sentiment Tree Dataset 8**

**5 Block diagram of the naïve bayes classifier 9**

**6 Dataset after dropping all the unwanted characters 9**

**7 Dataset after small casing all the characters 9**

**8 Dataset after dropping all the stop words 9**

**9 Dataset after stemming all the words 10**

**10 Visualizing sst-2 dataset 11**

**11 Confusion matrix of naïve bayes 11**

**12 Confusion matrix of SVM 11**

**13 Table representing the accuracies 11**

1. **LIBRARIES USED**

**Library Functioning**

*Pandas* : Python library used for working with data sets, in this

project pandas is used to read the dataset.

*nltk*  : NLTK is a natural language tool kit that provides

various natural language algorithms

*Porter stemmer* : A stemming algorithm provided by the natural

language tool kit that is used to stem the words

*Count vectorizer* : Library provided by the sklearn that is used to vectorize

the words based on the frequency count

*GaussianNB* : Architecture provided by sklearn which is built upon

the concept of naïve bayes

*Matplotlib*  : It is used for visualizing the data graphically

*re* : re represents the regular expression primarily used for

string matching and string manipulation

1. **INTRODUCTION**

In general, a text classification model either sentence level or a document level can identify whether the test is representing either of the following three

1. Positive Emotion (+1)
2. Negative Emotion (-1)
3. Neutral Emotion (0)

The expression could be easy to distinguish which include “Amazing customer service”, “Loved it”, … and some of them cannot be easily distinguishable(Hard expressions) by the machine which include “Not bad”, “Not good”, … so, our machine should be trained in such a way that it can distinguish even the hard expressions. There are different approaches to classify the sentence or a text which can include lexicon-based approach which uses pre-defined sentiment lexicon to score a document by aggregating the sentiment scores of all the word in the document, machine learning approach by considering it as a binary classification problem. The major step in the text classification is the converting the word into a vectorized form which can be done in many ways which can include bag of words method, frequency of occurrence method, word embedding, Bert method (Bi direction encoder using transformers). Among all the methods BERT method is the most popular method to vectorize the words, as the name suggests it is a bi direction encoder which means that it vectorize the word even by considering the neighbouring words in the sentence so that the sentence like “Not bad” can be classified as a positive, whereas the other encoding methods relies on the reptations and the probability of the word occurring in a labelled sentence.

In this project a machine learning based classification model has been designed using the concept of bayes theorem and SVM and finally the experimental results are compared.

1. **CONCEPTS USED**
2. **Bayes Theorm**

Bayes’ theorem describes the probability of occurrence of an event related to any condition. It is also considered for the case of conditional probability. Bayes theorem is also known as the formula for the probability of “causes”.

Let,

then, Bayes theorm states that if are the events associated with a sample space S, where the union of returns the sample space then,

1. **Stemming**

Stemming is one of the most important pre-processing step in text classification which converts every word into its root word, in order to make the data normalized and more generic. There are some inbuilt libraries and algorithms to achieve the stemming process where porter stemmer is one of them which is provided by natural language tool kit. The stemming of the sentence is as shown below

|  |  |
| --- | --- |
| **Original sentence** | The nature are very lovely but sometimes becomes dangerous too |
| **Stemmed sentence** | The natur are very love but sometime become danger to |

**Fig(1): Stemming of a sentence**

1. **Bag of Words**

This is one of the way of vectorizing the words which includes the following steps:

Step1: Generate the tokens

Step2: Create a table of words and their corresponding sentence labels

Let us take two sentences as shown, assuming that they are pre-processed before

|  |  |
| --- | --- |
| wow love place | 1 |
| crust not good | 0 |
| not tasti textur nasti | 0 |

**Step1**: Generating the tokens

The tokens are basically the unique words in the sentences which are as shown in the table below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| wow | love | place | crust | not | good | tasti | textur | nasti | Labels |

**Fig(2): Token Generation from sentences**

**Step2**: Creating table using the generated tokens in step1

The bag of words representation of the sentences will be as shown in the table below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| wow | love | place | crust | not | good | tasti | textur | nasti | Labels |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |

**Fig(3): Bag-of-words Representation**

**Datasets**

The dataset used for the project is Stanford Sentiment Tree bank (SST-2 dataset), a corpus with fully labelled parse trees that allows for a complete analysis of the compositional effects of the sentiment language. SSt-2 dataset consists of 11,855 single sentences extracted from the movie reviews.

The dataset will be as shown in the table below. The first column of the dataset consists of sentences and the second column consists of their corresponding sentiments. The dataset will be as shown in the table below

|  |  |
| --- | --- |
| Sentence | Label |
| Hide new secretions from the parental units | 0 |
| Contains no wit, only labored gags | 0 |
| That loves its charaters and communicates something rather beautiful about human nature | 1 |
| Remains utterly satisfied to remain the same throughput | 0 |
| On the worst revenge-of-the-nerds clich the filmmakers could dredge up | 0 |

**Fig(4): Stanford Sentiment Tree Dataset**

**VII ALGORITHM**

As it is discussed earlier the model uses the concept of naïve bayes, but before training the data set with the naïve bayes classifier the data is pre-processed as shown in the figure below without abstracting the data pre-processing steps.

**Diagram

Description automatically generatedFig(5): Block diagram of the naïve bayes classifier**

Let us take the top 5 sentences from the SST-2 dataset as mentioned in the figure() then the pre-processing steps will be as shown in the tables given below:

1. Dropping all the unwanted characters from the text.

|  |  |
| --- | --- |
| Sentence | Label |
| Hide new secretions from the parental units | 0 |
| Contains no wit only labored gags | 0 |
| That loves its charaters and communicates something rather beautiful about human nature | 1 |
| Remains utterly satisfied to remain the same throughput | 0 |
| On the worst revenge of the nerds clich the filmmakers could dredge up | 0 |

**Fig(6): Dataset after dropping all the unwanted characters**

1. Smaller casing all the words

|  |  |
| --- | --- |
| Sentence | Label |
| hide new secretions from the parental units | 0 |
| contains no wit only labored gags | 0 |
| that loves its charaters and communicates something rather beautiful about human nature | 1 |
| remains utterly satisfied to remain the same throughput | 0 |
| on the worst revenge of the nerds clich the filmmakers could dredge up | 0 |

**Fig(7): Dataset after small casing all the characters**

1. Removing all the stopwords

|  |  |
| --- | --- |
| Sentence | Label |
| hide new secretions parental units | 0 |
| contains no wit labored gags | 0 |
| loves charaters communicates something beautiful human nature | 1 |
| remains utterly satisfied remain same throughput | 0 |
| worst revenge nerds clich filmmakers dredge | 0 |

**Fig(8): Dataset after dropping all the stopwords**

1. Stemming the words

|  |  |
| --- | --- |
| Sentence | Label |
| hide new secret parent unit | 0 |
| contain no wit labour gag | 0 |
| love character communicat something beautiful human nature | 1 |
| remain utter satisfy remain same throughput | 0 |
| Worse revenge nerd clich filmmaker dredge | 0 |

**Fig(9): Dataset after stemming all the words**

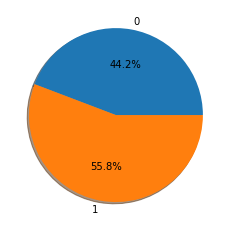
1. Vectorizing the words

Vectorizing of words can be done in various methods which include bag of words (BOW), dictionaries, based on frequencies, word embeddings and so on. In this project count vectorizer function provided by the scikit learn library which converts the text into vector form based on the frequency count of the word in the sentences.

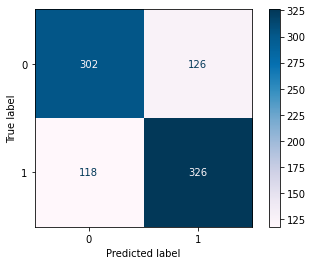
1. Train the naïve bayes classifier using the data generated in step4
2. Predict the outputs.

**VIII RESULTS**

The number of positives and negative sentiments in the Stanford sentiment tree dataset are as shown in the figure below



**Fig(10): Visualizing the SST-2 dataset**

****We ended up with getting an accuracy of 73% when trained with the gaussian naïve bayes classifier and 77% accuracy with the SVM classifier and the confusion matrices are build up on 872 reviews with using both SVM and naïve bayes model as shown in the fig() below:

**A picture containing text, electronics, screenshot

Description automatically generated**

**Fig(11): Confusion matrix of naïve bayes Fig(12): Confusion matrix of SVM**

The accuracies, precisions of the svm and naïve bayes classifiers are as shown in the table below:

|  |  |  |
| --- | --- | --- |
| **Classifier** | **Accuracy** | **Precision** |
| Naïve Bayes | **73** | **70.1** |
| SVM | **77** | **75** |

**Fig(13): Table representing the accuracies**

**IX CONCLUSION**

The naïve bayes is one of the classification techniques which is mostly used in sentiment classification like whether the mail is spam or ham, whether the review is positive or negative or neutral. But the main failure of the naïve bayes based sentiment classification is the word to vector conversion, as it just depends on the availability of the word in a particular sentence without checking with the neighbouring words or analysing the words correctly. So, we skip to the next level of conversion of word to vector representation which is by generation the word embeddings.

**X. REFERENCES**

1. Tutorial on naïve bayes classifier: <https://www.dataquest.io/blog/naive-bayes-tutorial/>

2. Dataset from Stanford sentiment treebank: [The Stanford Sentiment Treebank (SST2](https://nlp.stanford.edu/sentiment/index.html))

3. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context:

[1901.02860v3.pdf (arxiv.org)](https://arxiv.org/pdf/1901.02860v3.pdf)

1. Natural language tool kit documentation: [NLTK :: Natural Language Toolkit](https://www.nltk.org/)
2. Word2vec conversion tutorial**:** <https://youtu.be/UqRCEmrv1gQ>