

A Web Application of Text Sentiment Analysis.

Developed By:

Utsha Das

Roll: 143034

Department of Computer Science and Engineering

Rajshahi University of Engineering & Technology

Kazla, Rajshahi 6204

Email: utshadas5@gmail.com

Supervised By:

Bipro dip Pal

Assistant Professor

Department of Computer Science and Engineering

Rajshahi University of Engineering & Technology

Kazla, Rajshahi 6204

Email: biprodip.cse@gmail.com

I. INTRODUCTION

Sentiment Analysis is one of the greatest applications of Computer Science. Sentiment Analysis is an application of Natural Language processing which is used to identify the opinion of a person about the product or discussing topic ;whether it's negative, positive or neutral. It has a great importance in social media and business sector. In social media, it is used to measure the statistic. In business sector, marketing strategy of company, improvement of campaign success of a product, improvement of product messaging are greatly depend on it.

Our task to develop a web based application to determine the sensitivity of the user input text which is as comment. There are many well-known techniques to solve this problem. As a beginner, i have chosen **Naive Bayes classifier**. This technique is easy to understand and implement. The accuracy of this technique is not very good but enough to work with it.

II. METHODOLOGY

In this section, **Naive Bayes classifier** and it's application to sentiment analysis, **Text Parsing**, **Feature Vector** are discussed.

A. Naive Bayes Classifier

Naive Bayes Classifier is enough powerful algorithm for classification task. It is based on **Bayes Theorem**.

Bayes Theorem is a well-known in the field of probability. **Bayes Theorem** works on condition probability [1]. Conditional probability refers to the probability of happening of an event depends on the occurrence of another event which has already been.

If $B_1, B_2, B_3, \dots, B_n$ are mutually exclusive events of sample space S where

$$S = B_1 \cup B_2 \cup B_3 \cup \dots \cup B_n$$

and $P(B_i) > 0$; where $i=1, 2, 3, \dots, n$.

If A is another event of the sample space S such that $P(A) > 0$.

$$P(B_i|A) = \frac{P(B_i) \cdot P(A|B_i)}{\sum_{i=1}^n P(B_i) \cdot P(A|B_i)} \text{ where } i=1,2,3,\dots,n.$$

This is the **Bayes Theorem**[2].

B. Text Parsing

In this sub-section, text parsing procedure of the web application is introduced.

This is done using a **HTML** web page, where user input his/her comment about the topic. When the user input his/her comment, one copy of the input text is passed to the Python application file to be analyzed. **Fig.1** shows the procedure of text parsing as a flow chart.

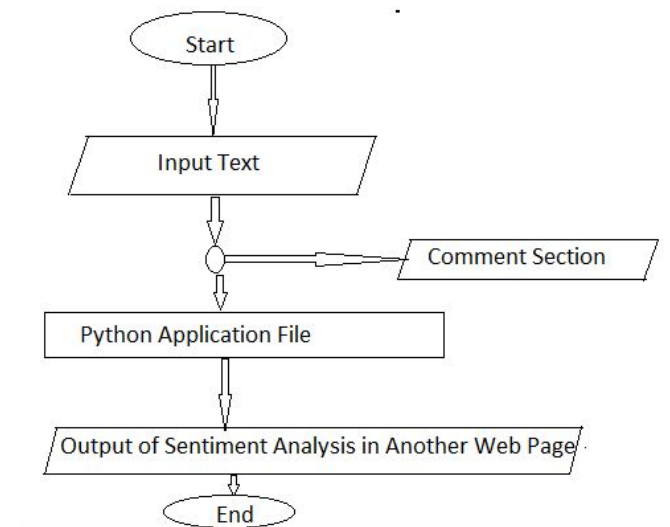


Fig. 1. Flow Chart of Text Parsing Technique used in the Web Application.

C. Feature Vector

Classification technique is completed using steps; training and prediction. For training phase, it requires training data. Then the training data is used to make predictions by the Classifier.

In my Web Application, positive, negative, neutral vocabulary are input as text file. Then **bag of words** model[3] is used to convert each word into feature. **Fig.2** shows the source of this.

Fig.3 shows a Snip of words featuring results.

Then the summation of these three feature sets makes the training set.

```
def word_feats(words):
    return dict([(word, True) for word in words])

positive_features = [(word_feats(pos), 'pos') for pos in positive_vocab]
negative_features = [(word_feats(neg), 'neg') for neg in negative_vocab]
neutral_features = [(word_feats(neu), 'neu') for neu in neutral_vocab]
```

Fig. 2. Source Code of word featuring using Bag of Words Model.

```
Positive Word Feature:
[('m': True, 'a': True, 'w': True, 'e': True, 'o': True, 'a': True), ('pos'), ({}, 'pos'),

Negative Word Feature:
[('d': True, 'a': True, 'b': True), ('neg'), ({}, 'neg'), ({'i': True, 'x': True, 'l': True

Neutral Word Feature:
[('i': True, 'm': True, 'v': True, 'e': True, 'o': True), ('neu'), ({'e': True, 't': True,
```

Fig. 3. A Snip of Results of Words Featuring using Bag of Words Model.

III. EXPERIMENTS

The goal of this project is to analyze the sentiment of user input comment. For this, i used **HTML** Web page to input the comment and Python as Programming Language.

A. Tools

- Web Page Designing Tools: **HTML5, CSS**
- Programming Language: **Python (version 3)**
- IDE: **PyCharm Community Edition 5.0.4**
- Server : **Local Server of Computer.**
- **NLTK**
- **Flask**

As the vocabulary set of positive word and negative word is too large, there are reduced. The reduced vocabulary sets are available here[4]

B. Results

From the view of user, input comment was used to determine the accuracy of the web application.

$$\text{Accuracy} = \frac{\text{No.ofCorrectResult}}{\text{No.ofTotalInput}} \times 100\%$$

No. of input changed and percentage of correction was recorded in the **Table 1**

No. of Total Input Comments	Accuracy(%)
10	60
20	68
30	63
40	65

TABLE I

EXPERIMENTAL DATA TABLE FOR NO. OF TOTAL INPUT COMMENTS AND ACCURACY(IN PERCENTAGE)

Fig.4, Fig.5, Fig.6, Fig.7 show the input and output result of the web application.

IV. CONCLUSION

In this paper, I have presented my own task to develop a web application to analyze the sentiment of user input comment. As beginner in this field, I had followed some tutorial series and Blog writings where are mentioned in acknowledgement section. The accuracy of this type of web application can be

Enter Your Comment:

He is abnormal.

Negative: 1.0Positive: 0.0

submit

Fig. 4. A Snip of Results of Input Comment.

Enter Your Comment:

beautiful

Negative: 0.0Positive: 1.0

submit

Fig. 5. A Snip of Results of Input Comment.

Enter Your Comment:

Welcome

Negative: 0.0Positive: 1.0

submit

Fig. 6. A Snip of Results of Input Comment.

Enter Your Comment:

I hate him.

Negative: 1.0Positive: 0.0

submit

Fig. 7. A Snip of Results of Input Comment.

developed using Neural Networking. My web application is enough helpful to analyze sentiment in small area. For type of Web Application can be further developed by using more powerful algorithm.

ACKNOWLEDGMENT

I have followed the website for web application development using Flask FrameWork of Python :

<https://pythonprogramming.net/practical-flask-introduction/>

I have also followed this website to understand the required algorithms:

<https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/>

REFERENCES

- [1] "Probability and Statistical Inference" by Robert V. Hogg, Elliot A.Tanis, Jagan Mohan Rao. Seventh Edition. Article 1.4, Page 35-37
- [2] "Probability and Statistical Inference" by Robert V. Hogg, Elliot A.Tanis, Jagan Mohan Rao. Seventh Edition. Article 1.6, Page 57-59
- [3] <https://pythonprogramminglanguage.com/bag-of-words/>
- [4] <https://drive.google.com/open?id=0B5oog5ywM0GcSlhZZmlySkIxVFU>