

Project Report

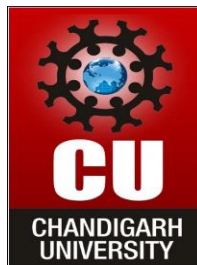
SENTIMENT ANALYSIS ON ARTICLES

Submitted for the requirement of

Project course

BACHELOR OF ENGINEERING

COMPUTER SCIENCE & ENGINEERING



Submitted to: Er. Charanpreet Kaur

Submitted By:

Tushar Lohani (19BCS1559)

Ankit Kumar (19BCS1561)

Sakshi Kumari (19BCS1570)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
CHANDIGARH UNIVERSITY, GHARUAN**

June 2021

CERTIFICATE

This is to certify that the work embodied in this Project Report entitled “**SENTIMENT ANALYSIS ON ARTICLES** ” being submitted by “**TUSHAR LOHANI,ANKIT KUMAR AND SAKSHI KUMARI** ” - UID “ **19BCS1559,19BCS1561 AND 19BCS1570** ” , 5th Semester for partial fulfillment of the requirement for the degree of “ **Bachelor of Engineering in Computer Science & Engineering** ” discipline in “ **Chandigarh University** ” during the academic session Aug-Nov 2021 is a record of bonafide piece of work, carried out by student under my supervision and guidance in the “ **Department of Computer Science & Engineering** ”, **Chandigarh University**.

APPROVED & GUIDED BY:

CHARANPREET KAUR

DECLARATION

We, students of Bachelor of Engineering in Computer Science & Engineering, 5th Semester, session: Aug – Nov 2021, Chandigarh University, hereby declare that the work presented in this Project Report entitled “SENTIMENT ANALYSIS ON ARTICLES” is the outcome of my own work, is bona fide and correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

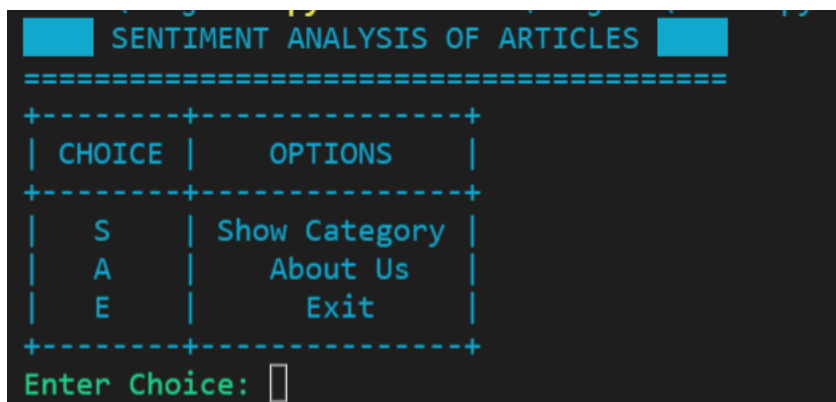
ABSTRACT

Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years. Its popularity is mainly due to two reasons. First, it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behaviors. Whenever we need to make a decision, we want to hear others' opinions. Second, it presents many challenging research problems, which had never been attempted before the year 2000. Part of the reason for the lack of study before was that there was little opinionated text in digital forms. It is thus no surprise that the inception and the rapid growth of the field coincide with those of the social media on the Web. In fact, the research has also spread outside of computer science to management sciences and social sciences due to its importance to business and society as a whole. In this talk, I will start with the discussion of the mainstream sentiment analysis research and then move on to describe some recent work on modeling comments, discussions, and debates, which represents another kind of analysis of sentiments and opinions.

Project Description

1. Introduction

This project is based on NLP that means Natural Language Processing. Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written -- referred to as natural language. It is a component of artificial intelligence. Sentiment analysis is the process of using natural language processing, text analysis, and statistics to analyze customer sentiment. The best businesses understand the sentiment of their customers - - what people are saying, how they're saying it, and what they mean. Customer sentiment can be found in tweets, comments, reviews, or other places where people mention your brand. Sentiment Analysis is the domain of understanding these emotions with software, and it's a must-understand for developers and business leaders in a modern workplace. As with many other fields, advances in deep learning have brought sentiment analysis into the foreground of cutting-edge algorithms. Today we use natural language processing, statistics, and text analysis to extract, and identify the sentiment of words into positive, negative. To address the context issue, a lot of research surrounding sentiment analysis has focused on feature engineering. Creating inputs to a model that recognize context, tone, and previous indications of sentiment can help increase accuracy and get a better overall sense of what the author is trying to say. For an interesting example, check out this paper in Knowledge-Based Systems that explores a framework for this kind of contextual focus. Search engines also use a similar technique called semantic search that determines the intent and contextual meaning of users' search terms.



```
SENTIMENT ANALYSIS OF ARTICLES
=====
+-----+-----+
| CHOICE | OPTIONS |
+-----+-----+
| S      | Show Category |
| A      | About Us      |
| E      | Exit          |
+-----+-----+
Enter Choice: 
```

Fig:- Command Based output of the project

2. Background

2.1 Objectives:

The principle targets of this task are:

1. To build up a sentiment analysis that can be used by everyone.
2. To make it cheap and easy to use.

2.2 Scope:

The extent of this project is:

1. To provide the use of technology.
2. To make an application free or cheap to use.
3. Easily understand able.

2.3 Level of Access:

The application will have two levels of access:

1. The Administrator
2. The Customer

2.4 Summary:

Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral. A sentiment analysis system for text analysis combines natural language processing (NLP) and machine learning techniques to assign weighted sentiment scores to the entities, topics, themes and categories within a sentence or phrase.

Sentiment analysis helps data analysts within large enterprises gauge public opinion, conduct nuanced market research, monitor brand and product reputation, and understand customer experiences. In addition, data analytics companies often integrate third-party Sentiment Analysis APIs into their own customer experience management, social media monitoring, or workforce analytics platform, in order to deliver useful insights to their own customers.

This article will explain how basic sentiment analysis works, evaluate the advantages and drawbacks of rules-based sentiment analysis, and outline the role of machine learning in sentiment analysis. Finally, we'll explore the top applications of sentiment analysis before concluding with some helpful resources for further learning.

3. Software and Hardware Requirements

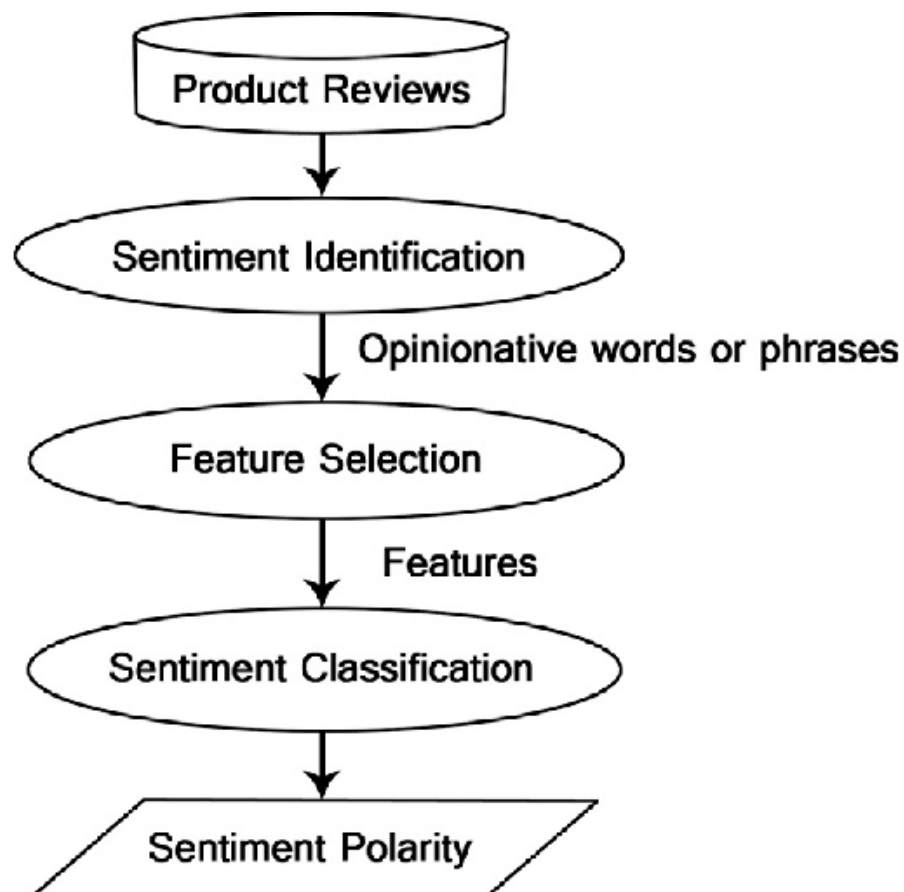
Using the requirement definition as a foundation, the requirements are divided into software and hardware. This is called system design.

3.1 Software requirements:

The programming language to use will be Python.

The project will make use of Visual studio code.

3.2 UML DIAGRAM:



3.3 PROJECT IMPLEMENTATION:

sentAnalysis.py:

```
from textblob import TextBlob
from webScrap import webScrapArticleText

mainURL = "https://asknkitkr.github.io/article/"
categories = [
    "categories/technology.html",
    "categories/book.html",
    "categories/entertainment.html",
    "categories/business.html",
    "categories/product.html",
]

def findSentiment(val):
    if val <= 0.1 and val >= -0.1:
        return "Neutral"
    elif val > 0.1:
        return "Positive"
    else:
        return "Negative"

def passSentimentList(url):
    sentiment = []
    for i in range(len(webScrapArticleText(url))):
        data = webScrapArticleText(url)[i]
        blob = TextBlob(data)
        sentiment.append(findSentiment(blob.sentiment.polarity))
        # polarity.append(blob.sentiment.polarity)
    return sentiment
```

webScrap.py

```
from bs4 import BeautifulSoup
import requests
```

```

from urllib.request import urlopen
from urllib.error import HTTPError, URLError

# To check the connection to the website
def checkURLConnection(url):
    msg = ""
    try:
        urlopen(url)
    except HTTPError as e: # HTTPError is a subclass of URLError
        msg = 'HTTP Error: ' + str(e.code)
    except URLError as e: # URLError is a subclass of OSError
        msg = 'URL Error: ' + str(e.reason)
    else: # No exception was raised
        msg = "Active"
    return msg

# To scrap the article titles from the website
def webScrapArticleTitle(url):
    titles_list = []
    html = requests.get(url)
    soup = BeautifulSoup(html.text, "lxml")
    articles = soup.find_all('div', class_='card')
    for index, article in enumerate(articles):
        title = article.find('div', class_='card-
header').text.replace(' ', '').replace('\n', ' ')
        titles_list.append(title)
        # print('{}. {}'.format(index+1, title.replace('\n', ' ')))
    return titles_list

# To scrap the article contents from the website
def webScrapArticleText(url):
    text_list = []
    html = requests.get(url)
    soup = BeautifulSoup(html.text, "lxml")
    articles = soup.find_all('div', class_='card')
    for index, article in enumerate(articles):
        text_articles = article.find('div', class_='card-
body').text.replace(' ', '').replace('\n', ' ')
        text_list.append(text_articles)

```

```
    # print(text_articles)
    return text_list
```

main2.py

```
# python import files
from sentAnalysis import passSentimentList
from webScrap import checkURLConnection, webScrapArticleTitle,
webScrapArticleText
from colorama import *
from prettytable import PrettyTable

# main link URL + category
mainURL = "https://asknkitkr.github.io/article/"
categories = [
    "categories/technology.html",
    "categories/book.html",
    "categories/entertainment.html",
    "categories/business.html",
    "categories/product.html",
]

MENU = {
    1: "Technology",
    2: "Book",
    3: "Entertainment",
    4: "Business",
    5: "Product",
}

init()

def menu():
    Menu = PrettyTable(["CHOICE", "OPTIONS"])
    Menu.add_row(["S", "Show Category"])
    Menu.add_row(["A", "About Us"])
    Menu.add_row(["E", "Exit"])
    print(Menu)
```

```

choice = input(Fore.LIGHTGREEN_EX + "Enter Choice: ")
if choice == "S" or choice == "s":
    showCategory()
elif choice == "A" or choice == "a":
    aboutUs()
elif choice == "E" or choice == "e":
    exit()
else:
    print(Fore.RED + "Invalid Choice")
    Menu()

def aboutUs():
    print(Fore.YELLOW)
    about = PrettyTable(["UID", "NAME", "WORK"])
    about.add_row(["19BCS1559", "Tushar Lohani", "Sentiment Analysis, Research"])
    about.add_row(["19BCS1561", "Ankit Kumar", "Web Scrapping, Sentiment Analysis"])
    about.add_row(["19BCS1570", "Sakshi Kumari", "Research, Documentation, Testing"])
    print(about)
    while(True):
        op = input("Press B to back: ")
        if op == 'B' or op == 'b':
            menu()
        else:
            print(Fore.RED + "Invalid Input")

def showCategory():
    print(Fore.YELLOW + "\nSELECT CATEGORY:")
    category = PrettyTable(["CHOICE", "OPTIONS"])
    for key, value in MENU.items():
        category.add_row([key, value])
    print(category)
    i=1
    choice = input(Fore.LIGHTGREEN_EX + "Enter Choice: ")
    getArticle(int(choice))

```

```

def getArticle(n):
    i = 1
    if(checkURLConnection(mainURL + categories[n-1])):
        print(Fore.GREEN + "\nSTATUS: CONNECTED\n")
    else:
        print(Fore.RED + "\nSTATUS: NOT CONNECTED\n")

    url = mainURL + categories[n-1]
    article_title = getArticles(url)
    for title in article_title:
        print(Fore.LIGHTCYAN_EX + str(i) + ". " + title)
        i += 1

    select_article = int(input(Fore.LIGHTGREEN_EX + "Enter Article
Number: "))
    if select_article == 1:
        print(Fore.LIGHTWHITE_EX + webScrapArticleText(url)[0])
        print(Fore.LIGHTYELLOW_EX + "\nSentiment of the Article: " +
str(passSentimentList(url)[0]))
    elif select_article == 2:
        print(Fore.LIGHTWHITE_EX + webScrapArticleText(url)[1])
        print(Fore.LIGHTYELLOW_EX + "\nSentiment of the Article: " +
str(passSentimentList(url)[1]))
    elif select_article == 3:
        print(Fore.LIGHTWHITE_EX + webScrapArticleText(url)[2])
        print(Fore.LIGHTYELLOW_EX + "\nSentiment of the Article: " +
str(passSentimentList(url)[2]))
    elif select_article == 4:
        print(Fore.LIGHTWHITE_EX + webScrapArticleText(url)[3])
        print(Fore.LIGHTYELLOW_EX + "\nSentiment of the Article: " +
str(passSentimentList(url)[3]))
    elif select_article == 5:
        print(Fore.LIGHTWHITE_EX + webScrapArticleText(url)[4])
        print(Fore.LIGHTYELLOW_EX + "\nSentiment of the Article: " +
str(passSentimentList(url)[4]))

def articleText(n):
    url = mainURL + categories[n-1]
    article_text = getArticleContent(url)
    for n in article_text:

```

```

        print(article_text[n])

def getArticles(url):
    article_title = webScrapArticleTitle(url)
    return article_title

def getArticleContent(url):
    article_text = webScrapArticleText(url)
    return article_text

print(Fore.CYAN + "██████ SENTIMENT ANALYSIS OF ARTICLES ██████")
print(Fore.CYAN + "=====")
menu()

```

3.5. Experimental Results and Discussion

In this section of the document we will be discussing the verification and testing of each software component. All problems will be described in detail and the solutions we made to solve these problems. In this section we will also discuss our overall results of the project and what we could have done to improve upon our project. Future work for this project will also be mentioned in this section of the document.

4.1 Thinking about Basic:

Here we think about all basic things like how we design and create an application all the requirements and all the basic needs and think we requires for the project.

4.2 Design Models and Mockups Designing:

The models and mockups help to ensure clarity in view of the project as well as how it works. Stakeholders are to sit through this process as drawings are created.

4.3 Application Creation:

The application was created using python programming language. It was coded in Visual studio code and functions were created and all the library was imported using packages provided in python.

4.4 Testing:

Testing will be implemented on software. Test cases may be used to guide and understand the basic actions of both customers and employees. Any bugs or errors that occur will be identified and resolved.

4.5 Finalization and Reports:

All testing and function processes are finalized at this stage. Reports will be created to ensure all information and functionality is clear in order to make the user manual and to help ensure employees can use the software with ease.

5.Conclusions:

In conclusion, the system will be able to serve as an application when it is finally developed, where these small upcoming companies can make use of it to publish their services in a wide range and also help the company to manage their service more effectively. On the other hand, it will enable customers to freely make their desire choice more freely and interactively. In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add home automation.
- We can give more advance software for virtual assistant including more facilities.
- We will host the platform on online servers to make it accessible worldwide.

6. Bibliography

Books Used:

1. Software Engineering - R.S.Pressman
2. MysqlBy McGrawhillPublication

Websites used:

1. Google for problemsolving
2. Quora
3. Youtube
4. Stackoverflow