www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 11, November 2024, pp : 28-31

e-ISSN : 2583-1062
Impact Factor : 7.001

HEALTH-INSIGHTS

A. Akshitha[1], A. Naveen Kumar[2], A. Sri Varun[3], G. Abhi[4], B. Abhishek kumar[5], Chinni krishna sir[6]

[1,2,3,4,5]B. Tech School of Engineering Computer Science – AI&ML Malla Reddy University, India.

[6]Guide: Asst Professor School of Engineering Computer Science – AI&ML Malla Reddy University, India.

ABSTRACT

Sentiment analysis is a technique used to classify the polarity of sentiments and opinions expressed in text data. This classification may be binary, distinguishing between positive and negative sentiments, or multiple, identifying emotions such as happiness, anger, sadness, and disgust. In this project, we aim to develop a sentiment analysis system using Python, a popular language for data science and machine learning. The sentiment analysis system will use machine learning techniques, including natural language processing and classification algorithms, to analyze text data and determine the sentiment scores. The system will be trained using a labeled dataset, which includes text data and their corresponding sentiment labels. The ultimate goal of this project is to develop a robust and accurate sentiment analysis system that can be applied to various applications, such as social media monitoring, customer feedback analysis, and market research. Additionally, this project will provide an opportunity to learn about machine learning techniques, including text preprocessing, feature extraction, and model selection, and apply them to realworld problems. In summary, this project aims to develop a sentiment analysis system using Python, which can classify the polarity of sentiments and opinions expressed in text data accurately. It will leverage machine learning techniques and can be applied to various applications, including social media monitoring, customer feedback analysis, and market research

Keywords: Python, Nlp

1. INTRODUCTION

Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative our neutral.it is

contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. However, analysis of social media streams is usually restricted to just basic sentiment analysis and count based metrics.

Competitive analysis: To compare the sentiment of reviews of a particular product or service against its competitors. By analyzing the sentiment of reviews, companies can identify

their strengths and weaknesses compared to their competitors.

Customer experience improvement: To improve the overall customer experience by analyzing feedback from customer reviews. By analyzing customer feedback, companies can identify pain points and make changes to improve the customer experience.

2. REQUIRED TOOLS

a) Software Requirements

Programming Language: Choose a programming language that is suitable for implementing the sentiment analysis algorithms and handling the data processing tasks. Popular choices include Python, Java, and R.

Development Environment: Set up an integrated development environment (IDE) to facilitate coding, debugging, and project management. Examples include PyCharm, Jupyter Notebook, Eclipse, or Visual Studio Code.

Natural Language Processing (NLP) Libraries: Utilize NLP libraries to process and analyze textual data. Commonly used libraries for sentiment analysis include NLTK (Natural Language Toolkit), spaCy, TextBlob, and Stanford NLP.

Machine Learning Libraries: If you are employing machine learning-based approaches, you will need relevant libraries for model training and evaluation. Popular libraries include scikitlearn, TensorFlow, Keras, and PyTorch.

Sentiment Lexicons: Depending on your project requirements, you may need sentiment lexicons or dictionaries. These lexicons associate words or phrases with sentiment scores. Examples include SentiWordNet, AFINN, VADER (Valence Aware Dictionary and sEntiment Reasoner), and NRC Emotion Lexicon.

b) Hardware Requirements

- CPU: A modern multicore processor (e.g., Intel Core i5 or above) is typically sufficient for running Sentiment analysis.
- GPU: Graphics Processing Units (GPUs) can significantly accelerate the performance of sentiment analysis tasks, especially when utilizing deep learning algorithms.

- RAM: Sufficient memory is crucial for efficient processing.
- Internet Connectivity: If your sentiment analysis

project involves data retrieval from online sources or the use of cloud-based services, a stable and reliable internet connection is necessary. .

- Storage: Sufficient storage space is required to store datasets, models, and application data.

3. MODULES

a) Natural Language Processing (NLP) Libraries:

A comprehensive library for NLP tasks, including tokenization, stemming, lemmatization, and sentiment analysis.

b) Machine Learning Libraries:

PyTorch: Another widely used deep learning library that provides flexible tools for sentiment analysis, including building and training neural networks.

Keras: A high-level neural networks API that runs on top of TensorFlow, simplifying the process of building and training sentiment analysis models.

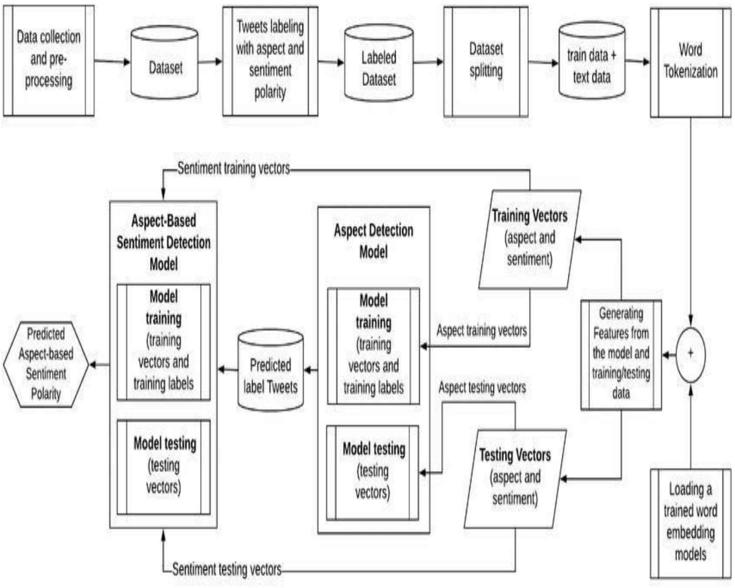
c) Data Processing and Analysis:

pandas: A powerful library for data manipulation and analysis, often used for preprocessing and organizing data for sentiment analysis.

NumPy: A fundamental library for numerical computing, useful for handling arrays and performing mathematical operations

4. ARCHITECTURE

The architecture of a sentiment analysis system typically involves several components working together to analyze and classify the sentiment of textual data. Here is a high-level architecture for sentiment analysis:


Data collection and pre-processing
Dataset
Tweets labeling with aspect and sentiment polarity
Labeled Dataset
Dataset splitting
train data + text data
Word Tokenization
Sentiment training vectors
Aspect-Based Sentiment Detection Model
Model training (training vectors and training labels
Model testing (testing vectors)
Predicted Aspect-based Sentiment Polarity
Predicted label Tweets
Aspect Detection Model
Model training (training vectors and training labels
Model testing (testing vectors)
Training Vectors (aspect and sentiment)
Aspect training vectors
Aspect testing vectors
Generating Features from the model and training/testing data
+
Testing Vectors (aspect and sentiment)
Loading a trained word embedding models
Sentiment testing vectors

Here are the main components of the architecture of a
sentiment analysis::

a) Data Collection: Gather textual data from various sources such as social media platforms,customer reviews,or news articles.Use APIs or web scraping techniques to collect the data .

b) Feature Extraction : Convert the preprocessed text into numerical features that can be used for sentiment analysis.

c) Sentiment Analysis Model:Train or utilize a

pretrained sentiment analysis model to predict the sentiment of the text.

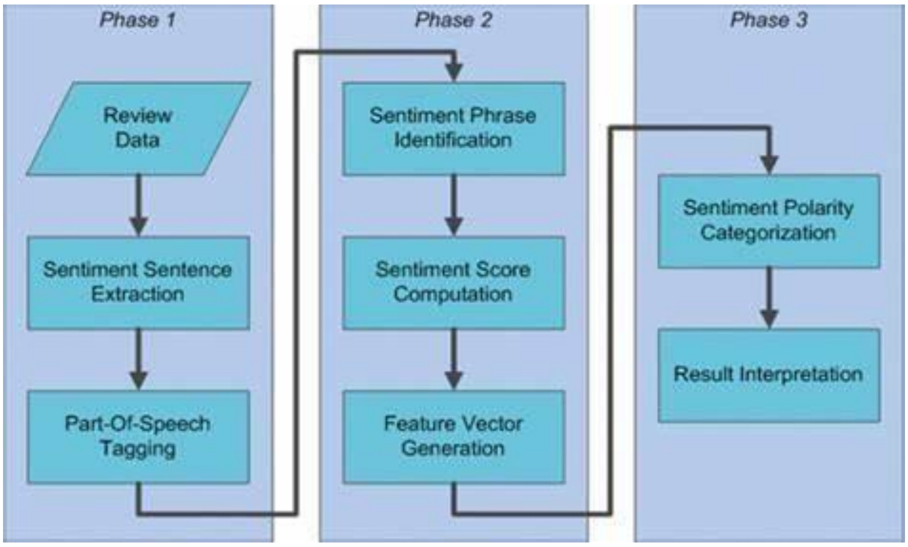
d) Sentiment Classification :Apply the trained model to features vectors to classify the sentiment of the text. The classification can be binary or multi-class or sentiment labels like happy, sad, angry etc.

e) Post-processing and Analysis: Evaluate and interpret the sentiment analysis results.Calculate sentiment scores or confidence levels for the classified

sentiment.

5. DESIGN

UML/ USECASE DIAGRAM:


Phase 1
Review Data
Sentiment Sentence Extraction
Part-Of-Speech Tagging
Phase 2
Sentiment Phrase Identification
Sentiment Score Computation
Feature Vector Generation
Phase 3
Sentiment Polarity Categorization
Result Interpretation

a) Define Project Scope: Clearly define the scope and objectives of your sentiment analysis project. Determine the specific types of sentiment you want to analyze (e.g., positive/negative, emotionbased), the target domain or language, and the intended application (e.g., social media monitoring, customer feedback analysis).

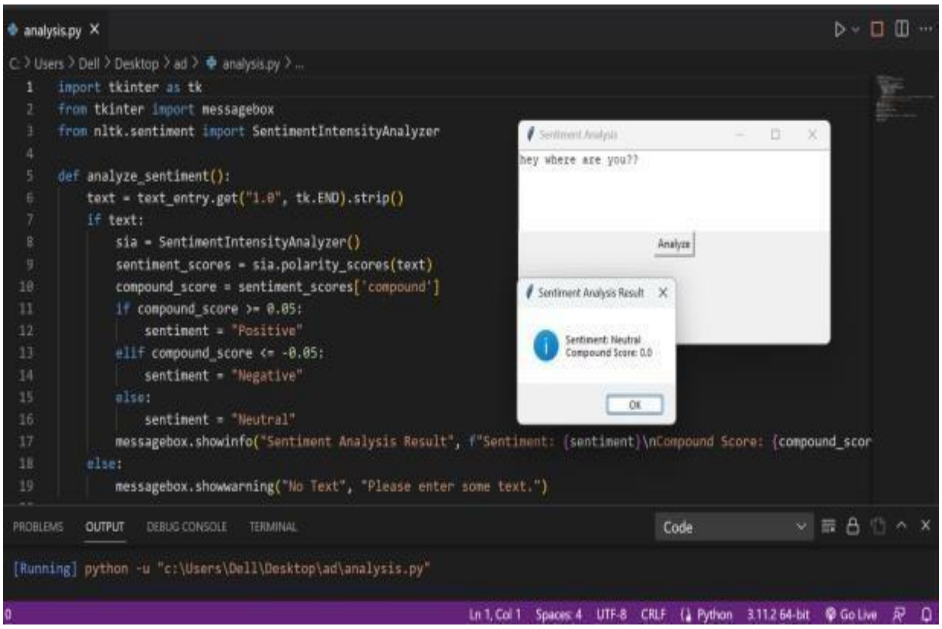
b) Data Collection and Labeling : Identify and collect relevant data for training and testing your sentiment analysis model.this may involve web scraping, accessing APIs, or using pre-existing labelled datasets.

c) Preprocessing and Feature Engineering: Clean and preprocess the collected text data by removing noise, special characters, and irrelevant information. Tokenize the text into words or phrases and apply techniques like stop word removal, stemming, or lemmatization to normalize the text.

d) Model Selection and Training: The system needs a database of facial images to compare against. It is important to ensure that the data collected is representative of the population and that the data is stored securely and ethically.

6. RESULTS

Executing the above code after importing the necessary modules and packages we get the following output:



```python
import tkinter as tk
from tkinter import messagebox
from nltk.sentiment import SentimentIntensityAnalyzer

def analyze_sentiment():
    text = text_entry.get("1.0", tk.END).strip()
    if text:
        sia = SentimentIntensityAnalyzer()
        sentiment_scores = sia.polarity_scores(text)
        compound_score = sentiment_scores['compound']
        if compound_score >= 0.05:
            sentiment = "Positive"
        elif compound_score <= -0.05:
            sentiment = "Negative"
        else:
            sentiment = "Neutral"
        messagebox.showinfo("Sentiment Analysis Result", f"Sentiment: {sentiment}\nCompound Score: {compound_scor
    else:
        messagebox.showwarning("No Text", "Please enter some text.")
```

```
hey where are you??
```

```
Sentiment Analysis Result
Sentiment: Neutral
Compound Score: 0.0
OK
```

```
[Running] python -u "c:\Users\Dell\Desktop\ad\analysis.py"
```

This is a screen shot of the output of the sentiment analysis showing the behaviour or state of the customer.

7. CONCLUSION

In conclusion, By analyzing sentiment, we can uncover patterns, trends, and sentiments hidden within textual data, allowing businesses and organizations to make informed decisions, tailor their products and services to customer preferences, and enhance customer satisfaction. Sentiment analysis also plays a vital role in social listening, enabling individuals and companies to understand public sentiment, detect emerging trends, and respond effectively to customer needs and concerns.

As sentiment analysis continues to evolve, it is essential to consider ethical implications, ensuring privacy, fairness, and transparency in the use of user-generated data. Additionally, ongoing research and development in the field will lead to more sophisticated models, enabling deeper understanding and more nuanced sentiment analysis.

In conclusion, sentiment analysis is a rapidly growing field that offers valuable insights into human sentiment and behavior, revolutionizing the way we understand and interact with textual data. By leveraging sentiment analysis effectively, individuals, businesses, and organizations can gain a competitive edge, make data-driven decisions, and foster stronger connections with their target audience.

ACKNOWLEDGMENT

My Major project would not have been successful without the help of several people. I would like to thank the personalities who were part pf my major project in numerous ways, those who gave us outstanding support from the beginning of the project. I am extremely thankful to our honorable pro-vice chancellor,VSK Reddy sir for providing necessary infrastructure and resources for the accomplishments of my project . I highly indebted to Prof.Kailasa Rao sir Dean, School Of

Engineering,for his support during the tenure of the project. I am very much obliged to our beloved Prof.Thayyaba Khatoon, Head od the Department of Artificial Intelligence & Machine Learning for providing the opportunity to undertake this project and encoutagement in completion of this project . I here by wish to express our deep sense of gratitude to Prof.Balaiah for the esteemed guidance , moral support and invaluable advice provided by them for the success of the project .

8. REFERENCES

[1] Our project Sentimental Analysis code is suggested by chatGpt and also we have made some changes in the code .We have implemented using popular libraries such as NLTK and scikitlearn in Python.

[2] This code uses the SentimentIntensityAnalyzer from the NLTK library to analyze the sentiment of a given sentence. It assigns sentiment scores to the sentence and interprets them to determine whether the sentiment is positive, negative, or neutral.

[3] We also used Tkinter in our code which provides a wide range of widgets and options for building interactive GUI applications. ScrolledText widget, which provides a text area that can handle longer texts.

[4] Users can enter or paste text into the widget. When the "Analyze" button is clicked, the analyze_sentiment function is called. Inside this function, sentiment analysis is performed on the text extracted from the ScrolledText widget using the SentimentIntensityAnalyzer from the NLTK library. The sentiment result is then displayed in a message box using messagebox.showinfo().