WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Sentiment Analysis using NLTK and FLASK

Mr. Rohan Sudhakar Jadhav[1], Mrs. Dipali Bhusari[2]

[1] Department of MCA , Trinity Academy Of Engineering , Pune , India,
[2]Assistant Professor of MCA , Trinity Academy Of Engineering , Pune , India

ABSTRACT

This research paper presents a web-based sentiment analysis system built using Natural Language Toolkit (NLTK) for natural language processing and Flask for deployment. The system automatically detects the sentiment (positive, negative, or neutral) of a given text, such as tweets, reviews, or comments. Sentiment analysis plays a vital role in understanding public opinion, improving customer service, and monitoring brand reputation. This project focuses on preprocessing textual data, using a Naïve Bayes classifier, and integrating the model into a lightweight and responsive Flask web application. The system aims to provide a real-time, accurate, and scalable solution for sentiment evaluation in diverse application domains.

.***KEYWORDS :*** *Sentiment Analysis, NLTK, Flask, Naïve Bayes Classifier, Natural Language Processing, Web Application*

INTRODUCTION

In today's digital world, individuals express opinions on social media platforms, forums, e-commerce websites, and news portals. Extracting and understanding the emotions behind these opinions is essential for businesses, governments, and individuals. Sentiment analysis, also known as opinion mining, refers to the use of natural language processing (NLP), text analysis, and computational linguistics to identify and extract subjective information from source text.

This paper presents a real-time web-based sentiment analysis application using Python's **NLTK library** and the **Flask web framework**. NLTK provides a suite of tools for text processing, including tokenization, stop-word removal, stemming, and classification, while Flask helps in developing and deploying the system on the web. The system allows users to input any sentence and get instant feedback on its sentiment, offering practical applications in marketing analysis, customer relationship management, and social media tracking.

LITREATURE SURVEY/BACKGROUND

- Sentiment analysis has evolved significantly in the past two decades. Early work by **Pang and Lee (2002)** applied machine learning classifiers like SVM and Naïve Bayes to sentiment-labeled movie reviews. Their work established the foundation for using supervised learning for sentiment classification.
- Later, **Go et al. (2009)** introduced the use of distant supervision by using emoticons in Twitter posts as labels for training classifiers, making it possible to build large datasets for training sentiment models. The approach of using pre-annotated corpora led to significant performance gains.
- **NLTK**, developed by Steven Bird and Edward Loper, has become a widely-used platform for building NLP models in education and research. It supports a wide range of preprocessing and machine learning tasks, and its integration with Naïve Bayes models makes it ideal for sentiment classification tasks.
- **Flask**, a micro web framework in Python, is commonly used for lightweight application deployment. It is minimal but extendable and works well for creating REST APIs or serving NLP-based models to the web.

These studies and technologies form the foundation of this paper's proposed system.

PROPOSED WORK/SYSTEM

1. System Overview:

The system is designed to classify the sentiment of a given text input. Users enter a sentence via a web form, and the backend processes it using the trained NLTK classifier. The output is the sentiment label (Positive, Negative, Neutral).

- Provide real-time sentiment classification
- Simplify the deployment of NLP tools
- Maintain a lightweight and scalable architecture

The proposed design ensures the system remains functional even with limited computational resources, making it suitable for educational and small-business use cases.

The core components of the system include:

 Input Layer: Users enter a sentence through a web-based form.

 Preprocessing Module: Text undergoes tokenization, stop-word removal, and stemming to prepare it for classification.

 Classifier Module: A trained **Naïve Bayes classifier** evaluates the sentiment of the processed text.

 Result Display: The predicted sentiment is shown to the user on the same webpage.

 Web Interface: Built using Flask, the interface is minimal, responsive, and easy to use.

2. System Architecture

The system architecture consists of five main stages:

 User Interface (Frontend)

- Accepts input from users via a web form
- Sends the input to the backend for processing

 Text Preprocessing (NLTK)

- **Tokenization**: Breaking text into words
- **Stop-word Removal**: Removing common filler words (e.g., “is”, “the”)
- **Stemming**: Converting words to their root form

 Feature Extraction

- Converts cleaned words into feature sets using a Bag-of-Words model
- Represents the text in a format understandable by classifiers

 Classification

- Uses Naïve Bayes to classify the sentiment
- Predicts one of three classes: Positive, Negative, or Neutral

 Result Display & Logging

- Returns the sentiment to the user
- Logs inputs and predictions for accuracy analysis

METHODOLOGY

 Data Collection

For model training, public sentiment-labeled datasets such as:

- **Movie Reviews Dataset (NLTK)**
- **Twitter Sentiment Corpus** are used. These datasets provide examples of both positive and negative texts.

 Preprocessing

- **Lowercasing**: Converts all characters to lowercase
- **Tokenizing**: Splits sentences into words

- **Removing Stopwords**: Eliminates common words that don't affect sentiment
- **Stemming/Lemmatization**: Reduces words to their base form

 Model Training

- A **Naïve Bayes Classifier** is trained using the feature vectors derived from the training data
- The model is saved for use during runtime

 Integration with Flask

- A Flask web server hosts the sentiment analysis model
- When a user inputs text, it calls the classifier and returns the sentiment

 Testing and Evaluation

- Accuracy is evaluated using standard metrics (precision, recall, F1-score)
- The system is tested with different input types (social media posts, reviews, etc.)

RESULT AND DISCUSSIONS

The implementation of the proposed sentiment analysis system yielded satisfactory results in terms of classification accuracy, responsiveness, and usability. The system was tested using a curated dataset comprising user-generated text such as product reviews, social media posts, and short comments. These texts were subjected to classification using the Naïve Bayes algorithm integrated within the NLTK library.

Empirical evaluation of the model indicated a classification accuracy exceeding 80%, which is consistent with established benchmarks for traditional machine learning methods in sentiment analysis. The use of standard NLP preprocessing techniques—such as tokenization, stop-word removal, and stemming—contributed significantly to reducing noise in the input data, thereby enhancing the performance of the classifier.

The system demonstrated robustness in handling diverse sentence structures and variations in user input, especially in cases of clear polar expressions. Furthermore, the integration of the model within a Flask web framework ensured low latency during real-time processing, making it suitable for applications requiring immediate feedback.

CONCLUSION

This project successfully demonstrates a lightweight, real-time **sentiment analysis web application** using Python's NLTK library and Flask framework. It classifies user-input text as Positive, Negative, or Neutral based on a trained Naïve Bayes model. The application is easy to deploy, use, and modify for future enhancements.

Despite limitations in handling nuanced language, the system performs well for standard sentiment classification tasks. Future upgrades could integrate deep learning models, handle multiple languages, and incorporate emoji or slang interpretation. This project thus serves as a practical prototype for NLP-based sentiment analysis systems.

REFERENCES

[1] Pang, B., & Lee, L. (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques."

[2] Go, A., Bhayani, R., & Huang, L. (2009). "Twitter Sentiment Classification using Distant Supervision."

[3] Bird, S., Klein, E., & Loper, E. (2009). "Natural Language Processing with Python." O'Reilly Media.

[4] NLTK Documentation: https://www.nltk.org/

[5] Flask Documentation: https://flask.palletsprojects.com/

[6] Liu, B. (2012). "Sentiment Analysis and Opinion Mining." Morgan & Claypool Publishers.