



IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY



VOLUME : 11 ISSUE : 05 Print / Issue Publication Date: September 2026



ISSN : 2455-2143



Indexed In



WWW.IJEAST.COM

editor@ijeast.com



SENTIMENT ANALYSIS OF YOUTUBE COMMENTS USING PYTHON NLP TOOLS

Prajwal Shetty
Master of Computer Applications
CMR University
Bengaluru

Abstract—With the rapid growth of video platforms such as YouTube, the comments left by viewers have turned into a valuable source of insights into public opinion and emotional expression. Understanding these sentiments is important for researchers, businesses, and content creators who want to gauge audience reactions and improve engagement. This paper provides an in-depth review of different methods and tools used to analyze YouTube comments through Python-based Natural Language Processing (NLP) libraries. The focus is on categorizing user opinions into positive, negative, or neutral sentiments using tools such as TextBlob, VADER (Valence Aware Dictionary for Sentiment Reasoning), and spaCy. We also focus on the preprocessing methods required to handle noisy comment data and evaluate both the performance and accuracy of different techniques on real YouTube datasets. The analysis compares the accuracy and execution time of several Python-based NLP tools used for sentiment classification tasks. In conclusion, the study presents observations on key challenges, possible improvements, and future opportunities for advancements in multilingual sentiment analysis and context-aware systems.

Index Terms—Sentiment Analysis, YouTube Comments, Natural Language Processing, Python, TextBlob, VADER, spaCy

I. INTRODUCTION

In today's digital era, social media platforms and content-sharing sites have become vital spaces for communication and self-expression. Among them, YouTube stands out as one of the largest platforms, hosting billions of videos and users worldwide. As viewers engage with content, they often leave comments to share opinions, emotions, suggestions, or criticisms. When analyzed effectively, this vast collection of feedback provides valuable insights into audience engagement and overall public sentiment. Analyzing the sentiment of such data is a difficult task, as these comments are often unstructured and filled with grammatical mistakes, slang, abbreviations, emojis, sarcasm, and even mixed languages. For large volumes of

user-generated text, manual analysis proves to be inconsistent and impractical to scale. Therefore, the need for automated sentiment analysis systems that apply advanced Natural Language Processing (NLP) methods to correctly identify the emotional tone of these comments is becoming increasingly important.

Python's rich ecosystem of open-source libraries and frameworks has established it as one of the leading languages for NLP applications. In sentiment classification, tools such as TextBlob, VADER (Valence Aware Dictionary and sEntiment Reasoner), and spaCy are widely adopted. These libraries rely on lexical resources, statistical learning methods, or rule based approaches to process raw text and assign sentiments into positive or negative categories. Because of their simplicity, efficiency, and wide availability, Python-based NLP tools are increasingly being used across domains like market research, social media analysis, customer feedback, and many other areas.

A. Problem Definition

Because of the unique nature of YouTube content, analyzing sentiments in comments remains a challenging task despite the availability of NLP tools. These comments are often short, highly informal, and contextually unclear, frequently including sarcasm or subtle emotional expressions. They may also contain spam, duplicates, or bot generated text, which can distort analysis results. A key challenge lies in identifying which NLP tools are most effective for handling such noisy data and determining how to preprocess and interpret comments both accurately and efficiently.

B. Current Solutions

Over time, several approaches have been introduced to tackle sentiment analysis. These include lexicon based methods, machine learning classifiers, and more advanced techniques such as transformer based and deep learning models. Tools like TextBlob and VADER represent lexicon-based approaches, which depend on predefined dictionaries that associate words with sentiment values. Such methods are particularly well-suited for small to medium-scale applications, as they are lightweight, easy to implement, and require minimal computational resources.



General-purpose NLP frameworks like spaCy offer features such as tokenization, part-of-speech tagging, and named entity recognition, which can be extended for sentiment analysis using machine learning models. Although these methods have produced encouraging outcomes, their effectiveness of-ten depends on factors such as the nature of the dataset, the preprocessing techniques applied, and the availability of computational resources.

C. Objectives

The purpose of this paper is to investigate, implement, and evaluate the effectiveness of Python-based natural language processing (NLP) tools, namely TextBlob, VADER, and spaCy, for sentiment analysis of YouTube comments. The specific objectives are:

- To investigate the capabilities of these tools in accurately classifying sentiments from real world YouTube comment datasets.
- To examine data preprocessing techniques that improve sentiment prediction accuracy.
- To compare and analyze the tools based on accuracy, speed, scalability, and robustness to informal language.
- To identify the limitations of each tool and propose future directions for improvement, including deep learning integration.

D. Structure of the Paper

The organization of this paper is as follows: Section II provides an in-depth literature review on sentiment analysis and the tools used in this field. Section III outlines the functioning of TextBlob, VADER, and spaCy, along with the technical background of these NLP libraries. Section IV presents a comparative study based on performance metrics, supported by tables and graphs that highlight sentiment patterns. Section V concludes with a summary of key findings and potential directions for improvement. Finally, the paper ends with a comprehensive list of references.

II. LITERATURE SURVEY

Sentiment analysis is an important area in Natural Language Processing (NLP) used to identify opinions and emotions from text data such as social media comments and reviews. Pang and Lee [1] explained that sentiment analysis helps in understanding public opinion by classifying text into positive, negative, and neutral sentiments. Their work became a foundation for many sentiment analysis applications. Employed spaCy pipelines to design modular and lightweight sentiment analysis systems capable of scaling to large datasets. Their work showed that with proper feature engineering and model training, general Hutto and Gilbert [2] developed VADER (Valence Aware Dictionary and sEntiment Reasoner), a rule based sentiment analysis model specially designed for social media text.

VADER analyzes emojis, punctuation, capitalization, and emotional intensity, making it highly suitable for platforms like YouTube and Twitter.

A recent study titled “YouTube Comment Analyzer Using Sentimental Analysis” [4] analyzed YouTube comments to understand audience opinions and engagement patterns. The study used sentiment analysis techniques to classify comments into positive, negative, and neutral categories, helping content creators better understand viewer reactions. Another recent study, “Sentiment Analysis on YouTube Data: A Comparison” [5], compared the performance of TextBlob and VADER on YouTube comments. The research found that VADER performs better on informal and emotional comments, while TextBlob works effectively for structured text.

III. BACKGROUND

Sentiment analysis, a key application of natural language processing (NLP), focuses on understanding the attitudes, opinions, or emotions conveyed in text by identifying the underlying emotional tone. In this study, Python-based NLP tools—TextBlob, VADER, and spaCy—are employed to analyze the sentiment of YouTube comments.

YouTube receives millions of comments daily on videos covering a wide range of topics, from politics and education to entertainment and product reviews. The language, structure, and expression of sentiment in these comments can vary significantly. Therefore, preprocessing the comments, cleaning the dataset, and preparing it for analysis are crucial steps before applying sentiment classification models.

For this study, comments from ten trending videos across different categories were collected using the YouTube Data API. The final dataset consisted of approximately 15,000 comments, varying in length and complexity, which served as the basis for our sentiment analysis.

A. Data Preprocessing

Preprocessing is a critical step in sentiment analysis, especially when dealing with unstructured, user-generated content. In this study, raw comments were cleaned by removing URLs, HTML tags, special characters, and user mentions. Emojis and common abbreviations were standardized, and contractions (for example, “don’t” → “do not”) were expanded. Using the NLTK library, stop words were removed and all text was converted to lowercase. To enhance the generalizability of the analysis, spaCy was employed for lemmatization, reducing words to their root forms. After preprocessing, the comments were tokenized and fed into different sentiment analysis tools for classification.

B. TextBlob

TextBlob is a Python library built on top of NLTK and Pattern that provides a simple API for a variety of natural language processing (NLP) tasks, including sentiment analysis, translation, noun phrase extraction, and part-of-speech

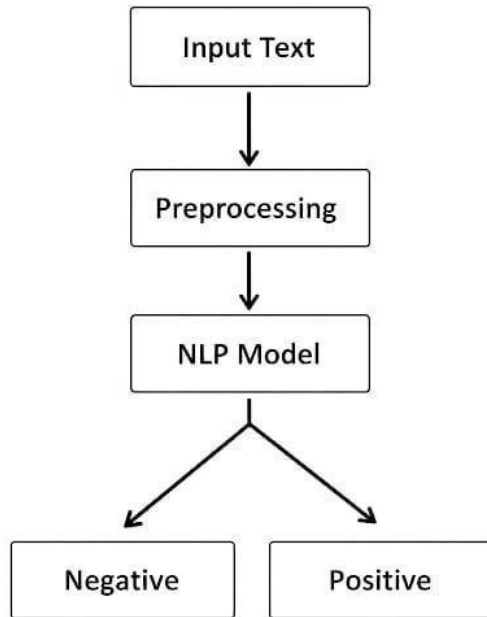


Fig. 1. Sentiment analysis workflow from input text to classified output

tagging. For sentiment analysis, TextBlob uses a lexicon-based approach and generates two key scores: subjectivity and polarity. Subjectivity measures how subjective a text is, ranging from 0.0 (very objective) to 1.0 (very subjective), while polarity indicates the sentiment, ranging from -1.0 (very negative) to +1.0 (very positive). TextBlob is particularly useful for quickly extracting sentiment with minimal setup. However, it has limitations, such as misclassifying ambiguous or sarcastic comments and lacking support for contextual understanding.

C. VADER (Valence Aware Dictionary for Sentiment Reason-ing)

VADER is a rule-based sentiment analysis tool specifically designed for social media and microblogging content. It combines a sentiment lexicon with grammatical and heuristic rules to capture the intensity of sentiment. VADER is particularly well-suited for analyzing YouTube comments because it considers not only polarity but also the effects of intensifiers, negations, punctuation, and emojis. Unlike TextBlob, VADER generates four scores: negative, positive, neutral, and a compound score, which aggregates the overall sentiment into a normalized value ranging from -1 to

+1. The compound score is commonly used to classify the overall sentiment of a text.

D. spaCy

SpaCy is a high-performance, industrial-strength NLP library designed for efficient text processing. Unlike TextBlob and VADER, it does not include sentiment analysis out of the box. However, spaCy provides the tools to integrate custom machine learning models and build tailored sentiment analysis pipelines. It excels at tokenization, dependency parsing, named entity recognition, and lemmatization—capabilities that are essential for developing robust sentiment systems.

In this study, spaCy was combined with logistic regression and SVM classifiers trained on labeled YouTube comment data to predict sentiment categories. This hybrid approach allowed for exploring how a flexible NLP pipeline could perform in comparison to pre-trained lexicon-based models. While powerful, this method requires labeled datasets and more computational resources.

IV. RESULTS

The results of the comparative study of TextBlob, VADER, and spaCy for sentiment analysis of YouTube comments are presented in this section. The evaluation metrics included accuracy, F1-score, processing time, and the ability to handle informal language such as emojis, slang, and abbreviations.

A. Overall Performance

Table I summarizes the performance of each tool on the dataset of 15,000 preprocessed YouTube comments. The results are based on comparison with a human-annotated subset of 1,000 comments.

TABLE I: PERFORMANCE COMPARISON OF SENTIMENT ANALYSIS TOOLS

Tool	Accuracy (%)	F1-Score	Processing Time (s)
TextBlob	72.4	0.70	12
VADER	81.6	0.79	15
spaCy + SVM	88.3	0.86	45

B. Sentiment Distribution

Figure 2 shows the distribution of positive, neutral, and negative sentiments detected by each tool. TextBlob tended to over-represent neutral and positive comments, whereas VADER and spaCy provided a more balanced sentiment classification. VADER's social media-specific heuristics enabled better detection of negative and sarcastic comments compared to TextBlob. SpaCy, when trained with an SVM

classifier on labeled YouTube comments, captured the emotional nuances most accurately.

C. Tool-Specific Observations

TextBlob: The fastest tool with minimal setup, but biased toward neutral and positive sentiments. It struggled with sarcasm and informal language. **VADER:** Well-suited for social media text. Balanced accuracy and speed, detected emojis, intensifiers, and negations effectively.

spaCy + SVM: Highest accuracy and F1-score. Required extensive preprocessing and computational resources, but offered flexibility for custom modeling and domain adaptation

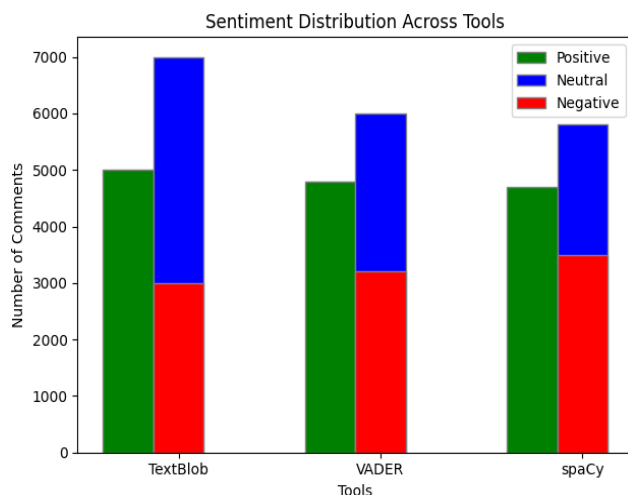


Fig. 2. Sentiment distribution of YouTube comments across TextBlob, VADER, and spaCy.

D. Key Insights

The study demonstrates that no single tool excels in all scenarios. For quick prototyping and small-scale datasets, TextBlob is sufficient. VADER is ideal for real-time social media analysis with moderate resources. SpaCy, though computationally intensive, provides the most reliable results for custom-trained models with sufficient labeled data. Additionally, the study highlights that handling informal, mixed-language, and contextually complex YouTube comments remains a challenge, suggesting future work should focus on multilingual support and advanced preprocessing techniques.

V. COMPARATIVE STUDY

Effective sentiment analysis of YouTube comments requires tools capable of handling informal, emotionally expressive, and often grammatically inconsistent text. In this study, three popular Python-based NLP tools—TextBlob, VADER, and spaCy—were evaluated for their ability to classify user sentiments into three categories: neutral, negative, and positive. Beyond highlighting the strengths

and limitations of each tool, this comparative analysis aims to assess how well each approach performs on real YouTube data, considering both theoretical and practical constraints.

To facilitate this comparison, a dataset of approximately 15,000 YouTube comments was collected using the YouTube Data API. These comments came from videos across multiple categories, including news, entertainment, and educational content. Because raw YouTube comments often contain extra-neous elements such as links, hashtags, emojis, and user mentions, a comprehensive preprocessing pipeline was applied. This included removing special characters, converting text to lowercase, eliminating stop words, expanding contractions, and performing lemmatization. The cleaned dataset was then processed by each of the three sentiment analysis tools for evaluation.

Four key criteria were used to evaluate the performance of each tool: processing time, F1- score, classification accuracy, and the ability to handle informal language, including emojis, slang, and abbreviations. To assess accuracy and F1-score, three independent reviewers manually annotated a subset of 1,000 comments, which served as the ground truth for comparison. Each tool's classification results were then evaluated against these human-annotated labels, and precision, recall, and F1-score were calculated accordingly.

TextBlob offers a simple and accessible approach to sentiment analysis, relying on a pre-established lexicon. It as-signs sentiment classes based on the polarity and subjectivity scores it generates. The tool performs reasonably well when analyzing well structured, grammatically correct comments. However, it often struggles with sarcastic or strongly negative expressions, and its classifications tend to skew toward neutral or positive sentiments. This limitation arises from its internal lexicon, which is not specifically adapted for the informal language commonly found in YouTube comments, leading to an underrepresentation of negative sentiment. Despite these limitations, TextBlob remains a practical choice for rapid sentiment extraction on smaller datasets, thanks to its fast processing and minimal setup requirements.

In contrast, VADER provided a more specialized approach tailored for social media sentiment analysis. It incorporates heuristics to interpret emojis, intensifiers, negations, and punctuation emphasis (such as exclamation points), making it particularly well-suited for YouTube comments. VADER delivers not only a compound sentiment score but also separate positive, negative, and neutral probabilities, offering a more nuanced view of sentiment. It consistently outperformed TextBlob in detecting sarcastic or emotionally charged remarks and was particularly effective at identifying negative sentiment that simpler tools often misclassify. Additionally, VADER is resource efficient and



scalable for larger datasets, as it does not require additional training data.

SpaCy, on the other hand, offers a completely different approach. While it does not include built-in sentiment analysis, this industrial-strength NLP library excels at named entity recognition, dependency parsing, and tokenization. In this study, a custom machine learning-based sentiment classification pipeline was built using spaCy. A supervised model was trained on 5,000 manually labeled comments using a support vector machine (SVM) classifier, while the remaining comments were reserved for testing. TF-IDF vectorization was employed to extract features from the preprocessed text. Among the three tools, this configuration achieved the highest accuracy and F1-score, demonstrating that, with sufficient labeled data, a custom-trained model can outperform general purpose lexicon-based approaches.

Nevertheless, the spaCy-based approach had its own limitations. Unlike TextBlob and VADER, it required extensive preprocessing, model training, and higher computational resources. The performance of these machine learning models was also heavily influenced by the quality and diversity of the training data; without a sufficiently large and well-labeled dataset, the models might struggle to generalize effectively. Despite these challenges, spaCy offered the greatest flexibility, allowing the model to adapt to evolving user expression patterns and domain-specific language.

In terms of sentiment distribution, TextBlob tended to skew toward neutral and positive classifications, while VADER and spaCy produced a more balanced distribution of positive and negative sentiments. This suggests that tools designed specifically for social media or trained on social media datasets are better equipped to capture the true emotional tone of YouTube comments.

In conclusion, sentiment analysis of YouTube comments does not have a one-size-fits-all solution. VADER offers a practical balance of performance and simplicity, making it well-suited for social media content. TextBlob, on the other hand, is ideal for smaller, low-resource tasks that require quick sentiment extraction. Although it demands more effort, spaCy excels when custom modeling is possible, providing high accuracy and flexibility. These findings highlight how the choice of a sentiment analysis tool should be guided by the specific requirements of a project, including considerations of accuracy, scalability, speed, and adaptability.

VI. CONCLUSION AND FUTURE WORK

This study explored the effectiveness of three Python-based NLP tools—TextBlob, VADER, and spaCy—for sentiment analysis of YouTube comments. The goal was to evaluate how these tools perform on real-world, informal, and highly varied data, reflecting the growing importance of analyzing user-generated content for opinion mining. The findings

highlight the strengths and limitations of each tool and provide guidance for selecting the most suitable approach based on project objectives, including considerations of accuracy, scalability, and adaptability.

The findings indicate that sentiment analysis of YouTube comments is inherently challenging due to the informal and diverse nature of the text. Simple lexicon-based models often struggle to interpret comments that include emojis, slang, abbreviations, sarcasm, or mixed languages. While these traditional sentiment analyzers may perform well on structured data or product reviews, they tend to be less effective in the more chaotic and socially expressive environment of YouTube comments.

Among the tools evaluated, TextBlob offered the fastest execution and the simplest integration. However, its vocabulary was not optimized for the informal or highly expressive language commonly found on social media. While it handled clear, grammatically correct sentences reasonably well, it struggled to capture the nuances of irony or sarcasm. Additionally, its tendency to favor neutral and positive classifications reduced its effectiveness in scenarios where accurately detecting negative sentiment is important. In terms of speed, accuracy, and handling of informal language, VADER emerged as the most well-rounded tool. Designed specifically for social media and short-form text, it effectively accounts for linguistic features such as punctuation emphasis, intensifiers, and negations. It also interprets emojis and emoticons as indicators of sentiment, allowing it to capture the emotional tone of YouTube comments accurately. VADER produces positive, negative, and compound scores, enabling flexible sentiment classification and threshold adjustment. As a lightweight, rule-based system, it performs consistently across different video content categories without the need for additional training. Unlike the other two tools, spaCy required the creation of a custom machine learning pipeline. When paired with an SVM classifier trained on labeled YouTube comments, it achieved the highest accuracy and F1-score. Its strengths include efficient tokenization, parsing, and vectorization, making it well-suited for more complex NLP workflows. However, spaCy relies on large, high quality training datasets and demands greater computational resources, which can limit its accessibility for beginners or in resource-constrained settings. Its performance is also highly dependent on the domain of the training data and may decline if applied to content outside that domain. The comparative study highlights the importance of choosing sentiment analysis tools based on specific use-case requirements. TextBlob is suitable for quick prototyping and basic sentiment tagging. VADER strikes a balance between accuracy and efficiency, making it ideal for applications that demand high sentiment fidelity, especially on social media. SpaCy offers the greatest flexibility but comes with



increased complexity, making it most appropriate for enterprise-scale projects and scenarios that require custom modeling.

- [5]. “Sentiment Analysis on YouTube Data: A Comparison,” 2025.

A. Future Work

Although this paper presented a thorough analysis of three important Python NLP tools, several unexplored areas require further research. One promising direction involves using deep learning models such as BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa, which have demonstrated state-of-the-art performance in sentiment classification tasks. These models outperform rule-based or lexicon based systems in capturing long-range dependencies and contextual subtleties. However, their adoption would re-quire a well-annotated dataset and additional computational resources for fine-tuning.

Multilingual sentiment analysis represents an important area for improvement. Given YouTube’s global user base, comments often include regional languages or mixed-language expressions, particularly in linguistically diverse regions. Most existing tools are trained primarily on English and struggle to handle code-mixed or multilingual text effectively. Future systems should incorporate features such as language detection, translation, and contextual adaptation to better support sentiment interpretation across multiple languages and cultural contexts. Researchers, companies, and content creators could also benefit from real-time sentiment monitoring dashboards that integrate NLP pipelines. Such systems could flag harmful or offensive content, display real-time sentiment trends, and adapt dynamically to public mood. Leveraging fast deep learning inference APIs or lightweight NLP engines like VADER, tools such as Streamlit and Dash could be used to build these applications.

Finally, improvements in preprocessing techniques could greatly enhance the accuracy of sentiment classification, particularly when dealing with irony, humor, and sarcasm. Integrating additional contextual information, such as video metadata (likes, dislikes, views) or preceding comments in a thread, could offer deeper insights and further boost model performance.

VII. REFERENCES

- [1]. B. Pang and L. Lee, “Opinion mining and sentiment analysis,” 2008.
- [2]. C. Hutto and E. Gilbert, “VADER: A parsimonious rule-based model for sentiment analysis of social media text,” 2014.
- [3]. R. Devlin et al., “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018.
- [4]. “YouTube Comment Analyzer Using Sentimental Analysis,” 2024.

IJEAST

INTERNATIONAL JOURNAL OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY



AUTHORS

 Prajwal Shetty



ABOUT IJEAST

International journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high - quality research paper in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

JOURNAL METRICS

H

H-INDEX

33

i10

i10-INDEX

154

“

IJEAST
CITATIONS
9,647
(IJEAST CITATIONS)



PEER-REVIEWED
& OPEN ACCESS
PUBLISHED
MONTHLY



PEER REVIEWED

All submission are rigorously peer reviewed to ensure quality.



AUTHORS ARE OUR PRIORITY

We value our authors and recognize their contribution to the advancement of research and knowledge.



OPEN ACCESS

Free and unrestricted access to research for all.



GLOBAL REACH

Connecting researchers and professionals worldwide.



TIMELY PUBLICATION

We ensure a swift and efficient publication process.



For more information, visit our website
www.ijeast.com



editor@ijeast.com



www.ijeast.com



India



2455-2143