Search...

Python Tutorial   Interview Questions   Python Quiz   Python Glossary   Python Projects   Practice Python   Da

Sentiment Analysis using VADER – Using Python

Last Updated : 11 Dec, 2024

Sentiment analysis involves determining the emotional tone behind a body of text. One popular tool for sentiment analysis is **VADER (Valence Aware Dictionary and sEntiment Reasoner)**, a lexicon and rule-based sentiment analysis tool specifically designed for texts that contain informal language like social media posts and reviews.

In this article, we'll explore how to use VADER in Python for sentiment analysis and break down the key components of the sentiment analysis output.

What is VADER?

VADER (Valence Aware Dictionary and sEntiment Reasoner) is designed to handle sentiments in social media text and informal language.

Unlike traditional [sentiment analysis](#) methods, VADER is tailored to detect sentiment from short pieces of text, such as tweets, product reviews, or any user-generated content that may contain slang, emojis, and abbreviations. It uses a pre-built lexicon of words associated with sentiment values and applies a set of rules to calculate sentiment scores.

How VADER Works?

VADER operates by analyzing the polarity of words and assigning a sentiment score to each word based on its emotional value. The tool then combines these scores into an overall sentiment score for the entire text, which consists of four components:

1. **Positive (pos)**: The proportion of the text that expresses a positive sentiment.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

Got It !

3. **Neutral (neu)**: The proportion of the text that is neutral or lacks clear sentiment.
4. **Compound**: The aggregated sentiment score that ranges from -1 (extremely negative) to +1 (extremely positive).

Sentiment Scores

VADER returns four key components in its sentiment analysis:

- **pos**: Percentage of text that is positive.
- **neg**: Percentage of text that is negative.
- **neu**: Percentage of text that is neutral.
- **compound**: The overall sentiment score (aggregated sentiment).

The **compound** score is the most important and is computed as a normalized value between -1 (most negative) and +1 (most positive). It summarizes the sentiment of the text:

- **Compound score > 0.05**: Positive sentiment
- **Compound score < -0.05**: Negative sentiment
- **Compound score between -0.05 and 0.05**: Neutral sentiment

Implementation: Sentiment Analysis using VADER

In this implementation, we will use the VADER (Valence Aware Dictionary and sEntiment Reasoner) tool for performing sentiment analysis on a given text.

1. Import **SentimentIntensityAnalyzer** from **vaderSentiment** to analyze text sentiment.
2. Define **sentiment_scores()** to input a sentence and calculate sentiment scores using VADER.
3. Create an instance of **SentimentIntensityAnalyzer** to compute sentiment scores.
4. Use **polarity_scores()** to generate a sentiment dictionary with components: pos, neg, neu, and compound.
5. Display the sentiment dictionary and percentages for negative, neutral, and

6. Classify the overall sentiment based on the compound score: positive,
   negative, or neutral.

Let's proceed with the implementation:

```python
# import SentimentIntensityAnalyzer class from
vaderSentiment.vaderSentiment module.
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Function to print sentiments of the sentence.
def sentiment_scores(sentence):

    # Create a SentimentIntensityAnalyzer object.
    sid_obj = SentimentIntensityAnalyzer()

    # polarity_scores method of SentimentIntensityAnalyzer object gives a
sentiment dictionary.
    # which contains pos, neg, neu, and compound scores.
    sentiment_dict = sid_obj.polarity_scores(sentence)

    print("Overall sentiment dictionary is : ", sentiment_dict)
    print("Sentence was rated as ", sentiment_dict['neg']*100, "%
Negative")
    print("Sentence was rated as ", sentiment_dict['neu']*100, "% Neutral")
    print("Sentence was rated as ", sentiment_dict['pos']*100, "%
Positive")

    print("Sentence Overall Rated As", end=" ")

    # Decide sentiment as positive, negative, or neutral
    if sentiment_dict['compound'] >= 0.05 :
        print("Positive")
    elif sentiment_dict['compound'] <= -0.05 :
        print("Negative")
    else :
        print("Neutral")

# Driver code to test the function
if __name__ == "__main__" :

    print("\n1st Statement:")
    sentence = "Geeks For Geeks is the best portal for computer science
engineering students."
    sentiment_scores(sentence)

    print("\n2nd Statement:")
    sentence = "Study is going on as usual."
    sentiment_scores(sentence)

    print("\n3rd Statement:")
    sentence = "I am very sad today."
    sentiment_scores(sentence)
```

We use cookies to ensure you have the best browsing experience on our website. By using our
site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

```
1st statement :
Overall sentiment dictionary is :  {'neg': 0.165, 'neu': 0.588, 'pos': 0.247, 'compound': 0.5267}
sentence was rated as  16.5 % Negative
sentence was rated as  58.8 % Neutral
sentence was rated as  24.7 % Positive
Sentence Overall Rated As Positive

2nd Statement :
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
sentence was rated as  0.0 % Negative
sentence was rated as  100.0 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Neutral

3rd Statement :
Overall sentiment dictionary is :  {'neg': 0.508, 'neu': 0.492, 'pos': 0.0, 'compound': -0.4767}
sentence was rated as  50.8 % Negative
sentence was rated as  49.2 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Negative
```

The output of the code results in sentiment dictionary and classify each sentence based on the compound score.

Conclusion

VADER provides an easy-to-use and efficient method for sentiment analysis, particularly suitable for texts with informal language, such as tweets, reviews, and comments. By using the compound score, you can easily determine the overall sentiment of a sentence, while the individual positive, negative, and neutral scores give you a breakdown of the sentiment expressed.

| Comment | More info | Advertise with us |

Next Article

Artificial intelligence vs Machine Learning vs Deep Learning

Similar Reads

Python Tutorial | Learn Python Programming Language

Python Tutorial â€" Python is one of the most popular programming languages. Itâ€™s simple to use, packed with features and supported by a wide range of libraries and frameworks. Its clean syntax makes it beginner-...