

[Courses](#)[Tutorials](#)[Practice](#)[Contests](#)[Search...](#)[Sign In](#)[Python Tutorial](#)[Interview Questions](#)[Python Quiz](#)[Python Glossary](#)[Python Projects](#)[Practice Python](#)[Data Science With P](#)

Next Article:

[Twitter
Sentiment](#)

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.

Lavender
Oatmilk Latte

[ORDER NOW](#)

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.
2. **Negative (neg)**: The proportion of the text that expresses a negative sentiment.
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 positive sentiments.
6. Classify the overall sentiment based on the compound score: positive, negative, or neutral.

Let's proceed with the implementation:

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

```
2 from
  vaderSentiment.vaderSentimen
  t import
  SentimentIntensityAnalyzer
3
4 # Function to print
  sentiments of the sentence.
5 def
  sentiment_scores(sentence):
6
7     # Create a
  SentimentIntensityAnalyzer
  object.
8     sid_obj =
  SentimentIntensityAnalyzer()
9
10    # polarity_scores method
  of
  SentimentIntensityAnalyzer
  object gives a sentiment
  dictionary.
11    # which contains pos,
  neg, neu, and compound
  scores.
12    sentiment_dict =
  sid_obj.polarity_scores(sent
  ence)
13
14    print("Overall sentiment
  dictionary is : ",
  sentiment_dict)
15    print("Sentence was
  rated as ",
  sentiment_dict['neg']*100,
  "% Negative")
16    print("Sentence was
  rated as ",
  sentiment_dict['neu']*100,
  "% Neutral")
17    print("Sentence was
  rated as ",
  sentiment_dict['pos']*100,
  "% Positive")
18
19    print("Sentence Overall
  Rated As", end=" ")
20
21    # Decide sentiment as
  positive, negative, or
  neutral
22    if
```

```

    sentiment_dict['compound']
    >= 0.05 :
23         print("Positive")
24     elif
    sentiment_dict['compound']
    <= -0.05 :
25         print("Negative")
26     else :
27         print("Neutral")
28
29     # Driver code to test the
    function
30     if __name__ == "__main__" :
31
32         print("\n1st
    Statement:")
33         sentence = "Geeks For
    Geeks is the best portal for
    computer science engineering
    students."
34
    sentiment_scores(sentence)
35
36         print("\n2nd
    Statement:")
37         sentence = "Study is
    going on as usual."
38
    sentiment_scores(sentence)
39
40         print("\n3rd
    Statement:")
41         sentence = "I am very
    sad today."
42
    sentiment_scores(sentence)

```

Output :

```

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

Twitter Sentiment
Analysis using
Python

Similar Reads

Python | Sentiment Analysis using VADER

Sentiment analysis involves determining the emotional tone behind a body of text. One popular tool for

4 min read

Twitter Sentiment Analysis using Python

This article covers the sentiment analysis of any topic by parsing the tweets fetched from Twitter using Python.

10 min read

Python - Sentiment Analysis using Affin

Afinn is the simplest yet popular lexicons used for sentiment analysis developed by Finn Årup Nielsen. It

2 min read

Tweet Sentiment Analysis Using Python

This article covers the sentiment analysis of by parsing the tweets fetched from Twitter using the streamlit

4 min read

Facebook Sentiment Analysis using python

This article is a Facebook sentiment analysis using Vader, nowadays many government institutions and companies

6 min read

Twitter Sentiment Analysis on Russia-Ukraine

The role of social media in public opinion has been profound and evident since it started gaining attention.

4 min read

Sentiment Analysis using CatBoost

Sentiment analysis is crucial for understanding the emotional tone behind text data, making it invaluable for

4 min read

Flipkart Reviews Sentiment Analysis using

Sentiment analysis is a NLP task used to determine the sentiment behind textual data. In context of product

4 min read

Sentiment Analysis using Fuzzy Logic

Sentiment analysis, also known as opinion mining, is a crucial area of natural language processing (NLP) that

7 min read

Sentiment Analysis of Hindi Text - Python

Sentiment Analysis for Indic Language: This article exhibits how to use the library VADER for doing the

3 min read
