Asingh

Apr 6 · 3 min read · Listen

Sentiment analysis project using Python and the Natural Language Toolkit (NLTK) library

> *"This was created using ChatGPT"*

Introduction

Sentiment analysis is a popular technique in natural language processing that involves analyzing text data to determine the emotional tone or sentiment of the text. This can be useful in a variety of applications, such as social media monitoring, customer feedback analysis, and market research. In this tutorial, we will show you how to create a sentiment analysis project using Python and the NLTK library.

Prerequisites

To follow this tutorial, you will need to have Python 3 installed on your computer, as well as the NLTK library. You can install NLTK using pip, the Python package manager, by running the following command in your terminal:

```
pip install nltk
```

You will also need to download the VADER lexicon, which is a pre-trained lexicon for sentiment analysis included in the NLTK library. To download the lexicon, run the following command in your Pytho

```
import nltk
nltk.download('vader_lexicon')
```

Step 1: Import the necessary libraries

First, we need to import the necessary libraries for our project. In addition to NLTK, we will also use the `pandas` library for data processing and the `matplotlib` library for data visualization. We can import these libraries using the following code:

```
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import pandas as pd
import matplotlib.pyplot as plt
```

Step 2: Load the data

Next, we need to load our data into our Python program. For this tutorial, we will use a dataset of movie reviews from the IMDb website, which is available on Kaggle here. You can download the dataset and save it as a CSV file in your local directory. Then, we can load the data using the `pandas` library:

```
data = pd.read_csv('IMDB Dataset.csv')
```

The dataset contains two columns: `review` and `sentiment`. The `review` column contains the text of the movie reviews, while the `sentiment` column contains the label indicating whether the review is positive or negative.

Step 3: Analyze the sentiment

Now that we have loaded our data, we can use the NLTK library to analyze the sentiment of each movie review. We will define a function `analyze_sentiment()` that

takes a text as input and returns the sentiment score using the `SentimentIntensityAnalyzer()` class from the `nltk.sentiment.vader` module:

```
def analyze_sentiment(text):
    """
    Returns the sentiment score of the given text using the VADER sentiment analyz
    """
    sia = SentimentIntensityAnalyzer()
    sentiment_scores = sia.polarity_scores(text)
    return sentiment_scores
```

The sentiment score is a dictionary with four values: `positive`, `negative`, `neutral`, and `compound`. The `compound` score is a normalized score between -1 and 1, where -1 is very negative, 0 is neutral, and 1 is very positive.

We can apply this function to each movie review in our dataset using the `apply()` method of the `pandas` library:

```
data['sentiment_score'] = data['review'].apply(analyze_sentiment)
```

This will add a new column `sentiment_score` to our dataset containing the sentiment scores for each review.

Step 4: Visualize the results

Finally, we can visualize the results of our sentiment analysis using the `matplotlib` library. For example, we can create a histogram of the `compound` scores to see the distribution of positive, neutral, and negative reviews in our dataset:

```
# Extract the compound scores from the sentiment scores column
compound_scores = data['sentiment_score'].apply(lambda x: x['compound'])

# Plot a histogram of the compound scores
plt.hist(compound_scores, bins=50)
```

```
plt.title('Sentiment Analysis Results')
plt.xlabel('Compound Score')
plt.ylabel('Frequency')
plt.show()
```

This will generate a histogram that shows the distribution of compound scores in our dataset. We can see that the majority of the reviews have a compound score between -0.2 and 0.8, indicating that they are mostly neutral or positive.

Conclusion

In this tutorial, we have shown you how to create a sentiment analysis project using Python and the NLTK library. We started by loading a dataset of movie reviews, then used NLTK to analyze the sentiment of each review and calculate a sentiment score. Finally, we visualized the results using the `matplotlib` library. You can use these techniques to perform sentiment analysis on any text data, such as customer reviews, social media posts, or news articles.