Web Scraping, Sentiment Analysis, and API Development for Amazon iPhone 12 Reviews

1. **Introduction**

This project aims to scrape product reviews for the iPhone 12 from Amazon, perform sentiment analysis on the reviews, and create APIs to retrieve the sentiment scores and reviews based on specific criteria. The project utilizes various Python libraries for web scraping, data analysis, and API development.

The project is divided into four main steps:

- Scraping product reviews from Amazon.
- Finding the most frequently used words (best keywords) and least frequent words (worst keywords).
- Performing sentiment analysis to classify reviews as **positive**, **negative**, or **neutral**.
- Creating APIs to retrieve reviews and analyse their sentiment scores.

2. **Web Scraping**

Libraries Used

- **Requests**: For sending HTTP requests to the server.
- **Beautiful Soup**: For parsing HTML and extracting data.
- **MySQL Connector**: For connecting to a MySQL database and storing scraped data.

Data Extraction Process

Using Beautiful Soup, I identified the relevant HTML elements and classes that contain the product reviews on the Amazon page for the iPhone 12. The scraping process involved:

1. Sending GET requests to the product review pages.
2. Parsing the HTML content to extract review titles, texts, storage sizes, colors, and verified purchase statuses.
3. Iterating through all available review pages to gather comprehensive data.

Storing Data in MySQL

The extracted reviews were stored in a MySQL database using MySQL Connector. A table was created with appropriate columns to hold the review data.

3. Keyword Analysis

Libraries Used

- **NLTK**: For natural language processing tasks.
- **Collections**: For counting word frequencies.
- **Stopwords**: To filter out common words that do not contribute to meaning (e.g., "the", "is").
- **Regular Expressions (re)**: For text manipulation.

Methodology

I analyzed the scraped reviews to identify the most frequently used and least frequently used keywords. The steps included:

1. Tokenizing the review texts.
2. Removing stopwords and non-alphanumeric characters.
3. Counting word frequencies using collections.Counter.

This analysis provided insights into common themes and sentiments expressed in the reviews.

4. Sentiment Analysis

Libraries Used

- **NLTK**: For natural language processing tasks.
- **SentimentIntensityAnalyzer**: To measure sentiment polarity.
- **Pandas**: For data manipulation.
- **Seaborn**: For data visualization.
- **MySQL Connector**: To fetch data from the database.

Data Processing

The scraped review data was extracted from MySQL and converted into a CSV format for further analysis. The SentimentIntensityAnalyzer() was employed to evaluate each review's sentiment polarity, categorizing them into positive, negative, or neutral sentiments.

Results and Conclusion

The sentiment analysis yielded the following scores:

- Positive Reviews: Score of 8.61
- Negative Reviews: Score of 1.3
- Neutral Reviews: Score of 24.07

The analysis concluded that most reviews were neutral in sentiment, indicating a balanced perspective among users regarding their experiences with the iPhone 12. The small proportion of negative reviews suggests that while some users had concerns, many were satisfied with their purchase.

5. API Development

Libraries Used

- **Flask**: For creating web APIs.
- **Jsonify**: To format responses as JSON.
- **Threading**: To run the Flask app in a separate thread.
- **MySQL Connector**: To interact with the MySQL database.
- **TextBlob**: For performing sentiment analysis through an API.

API Functionality

Two APIs were developed:

1. **Sentiment Analysis API**:
   - Accepts POST requests with review text and returns a sentiment score in JSON format.
2. **Review Retrieval API**:
   - Accepts GET requests with parameters for color, storage size, and verified purchase status to fetch relevant reviews from the database.

Testing the APIs Using Postman:

- Sent a POST request to http://127.0.0.1:5001/sentiment with a sample review text to receive a sentiment score.
- Sent a GET request to http://127.0.0.1:5001/reviews?color=Black&style_name=128GB&verified_purchase=Yes with parameters to retrieve specific reviews from MySQL.

6. Conclusion

This project successfully demonstrated the end-to-end process of web scraping product reviews from Amazon, conducting sentiment analysis on those reviews, and creating APIs for easy access to this information. The analysis revealed valuable insights into customer sentiments regarding the iPhone 12.

by Akash Sakhare