Trustpilot Review Scraper Explanation

This script is designed to scrape customer reviews from a specific company's page on Trustpilot and store them in a MongoDB database.

1. Scraping Process

The script works by iterating through the review pages of the target Trustpilot URL:

Target URL: It uses a base URL (e.g., https://www.trustpilot.com/review/partsofcanada.com) and appends page numbers to navigate through sequential pages of reviews (?page=1, ?page=2, etc.).

Fetching Pages: For each page URL, the requests library sends an HTTP GET request to retrieve the page's content. A User-Agent header is included to mimic a web browser.

Error Handling: It checks for HTTP errors (like 404) during the request. A 404 error is treated as an indicator that the end of available review pages has been reached, and the scraping process stops. Other request errors are caught to prevent the script from crashing but allow it to attempt the next page.

Parsing HTML: The content of the successfully fetched page is parsed using BeautifulSoup, which helps in navigating the HTML structure.

Extracting Reviews: The script looks for specific HTML elements (<article> tags with a particular class) that contain individual review blocks.

Data Extraction per Review: For each review block found, it extracts:

- The main review text.
- The star rating (by finding the <img> tag within the rating element and reading its alt attribute).
- The name of the reviewer.

Hyperlink Display: As requested, it also finds all <a> tags (hyperlinks) on the page and prints their URLs to the console.

Text Processing: Using NLTK, it breaks down the review text into sentences and then into individual words (tokens).

Structuring Data: The extracted text, rating, reviewer name, source URL, page number, and a scrape timestamp are compiled into a Python dictionary.

Storing in MongoDB:

Before inserting into the database, it checks if a review with the same text, reviewer name, and source URL already exists to avoid inserting duplicates on subsequent runs.

Unique reviews are inserted as documents into a specified MongoDB collection (partsofcanada_reviews).

Rate Limiting: A delay (time.sleep(3)) is included after processing each page to reduce the frequency of requests to the server, which is a good practice for web scraping.

2. Methodology

The script employs the following tools and techniques:

requests: For making HTTP requests to fetch web pages.

beautifulsoup4: For parsing the HTML content and navigating the DOM tree to find specific elements.

re (Regular Expressions): Used to find elements where class names might have dynamic parts (e.g., typography_body-l__.*) or to extract information from strings (like the rating from the alt text).

nltk (Natural Language Toolkit): Used for basic text processing, specifically sent_tokenize for splitting text into sentences and word_tokenize for splitting sentences into words.

pymongo: The Python driver for MongoDB, used to connect to the database and insert the scraped data.

time: Used to introduce delays between requests.

Pagination Handling: Simple loop and URL parameter modification to navigate through pages.

Basic Duplicate Checking: collection.find_one() is used to check for pre-existing documents matching key fields before inserting.

3. Assumptions

The script relies on the following assumptions:

Trustpilot HTML Structure Consistency: The most critical assumption is that the HTML structure of Trustpilot review pages (specifically the class names and tag hierarchy used to identify review blocks, text, ratings, and reviewer names) remains stable. If Trustpilot updates its website's front-end code and changes these classes, the script's selectors will need to be updated.

Standard Pagination: Trustpilot continues to use the ?page= query parameter for navigating between review pages.

Network Access: The machine running the script has reliable internet access and can connect to both trustpilot.com and the MongoDB cluster.

MongoDB Configuration: The provided MongoDB connection string and credentials are correct and have write access to the specified database and collection.

NLTK Data: The necessary NLTK data ('punkt') can be downloaded and accessed by the script.

End of Pages Indicator: A 404 HTTP response reliably signifies that there are no more review pages to scrape.

No Advanced Anti-Scraping: Trustpilot does not implement overly aggressive anti-scraping measures (like complex JavaScript challenges, frequent IP blocking, or CAPTCHAs that require more sophisticated handling than a simple User-Agent and delay).

Duplicate Check Sufficiency: The basic duplicate check (based on review text, reviewer name, and source URL) is sufficient for the user's needs. A more robust check might require a unique review ID if available on the page.