Wikipedia Link Scraper

Sample Output & Verification

This document presents sample command-line interactions and their corresponding outputs from the Python Wikipedia Link Scraper script. These examples demonstrate the script's core functionality, adherence to requirements, and robust error handling.

1. Successful Scrape: JSON Output Format

This sample illustrates a successful execution of the script, initiating a scrape from a specified Wikipedia article and outputting the results in JSON format.

Command Executed:

```
python SpeerAssessment.py "https://en.wikipedia.org/wiki/Canada" 2 --out json
```

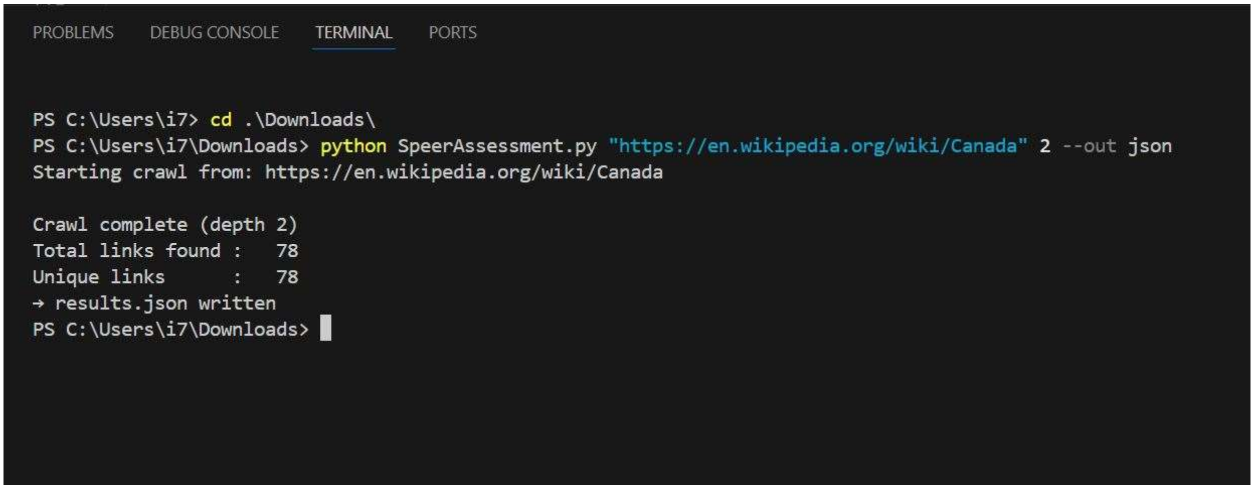
Description:

The script was executed to crawl the "Canada" Wikipedia page, delving 2 levels deep into interconnected links. The `--out json` argument directed the script to save the gathered unique links and metadata to a JSON file.

Expected Output:

Upon successful completion, the console output confirms the initiation of the crawl, the total number of links found, the count of unique links discovered within the specified depth, and a confirmation message indicating that `results.json` as per the provided script) has been successfully written.

Actual Output:



```
PS C:\Users\i7> cd .\Downloads\
PS C:\Users\i7\Downloads> python SpeerAssessment.py "https://en.wikipedia.org/wiki/Canada" 2 --out json
Starting crawl from: https://en.wikipedia.org/wiki/Canada

Crawl complete (depth 2)
Total links found :   78
Unique links      :   78
→ results.json written
PS C:\Users\i7\Downloads>
```

2. Successful Scrape: CSV Output Format

This sample demonstrates another successful execution, but with the output directed to a CSV file.

Command Executed:

```
python SpeerAssessment.py
"https://en.wikipedia.org/wiki/Python_(programming_language)" 3 --out csv
```

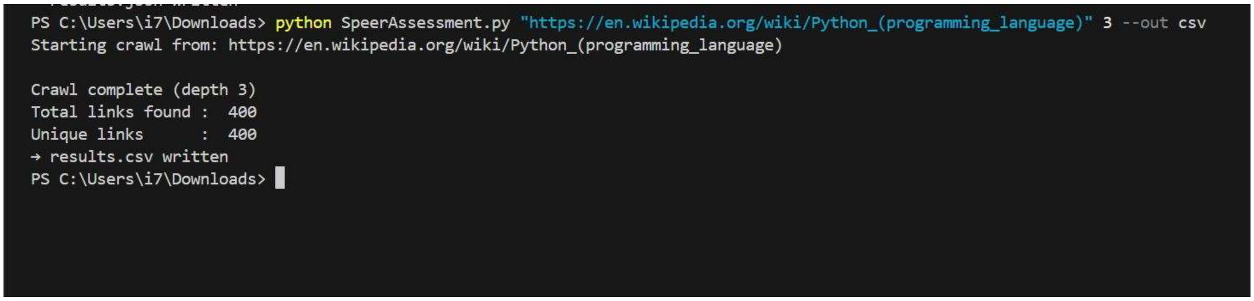
Description:

This execution targeted the "Python (programming language)" Wikipedia article, performing a deeper crawl of 3 levels. The `--out csv` argument ensured the results were stored in a CSV file, suitable for spreadsheet analysis.

Expected Output:

The console output confirms the start of the crawl, the successful completion up to 3 levels, the total and unique link counts found, and a message indicating that `results.csv` has been successfully written.

Actual Output:



```
PS C:\Users\i7\Downloads> python SpeerAssessment.py "https://en.wikipedia.org/wiki/Python_(programming_language)" 3 --out csv
Starting crawl from: https://en.wikipedia.org/wiki/Python_(programming_language)

Crawl complete (depth 3)
Total links found :  400
Unique links      :  400
→ results.csv written
PS C:\Users\i7\Downloads> █
```

3. Error Handling: Invalid Wikipedia Link

This sample demonstrates the script's validation capabilities when provided with an invalid initial Wikipedia link.

Command Executed:

```
python SpeerAssessment.py "https://google.com" 1 --out json
```

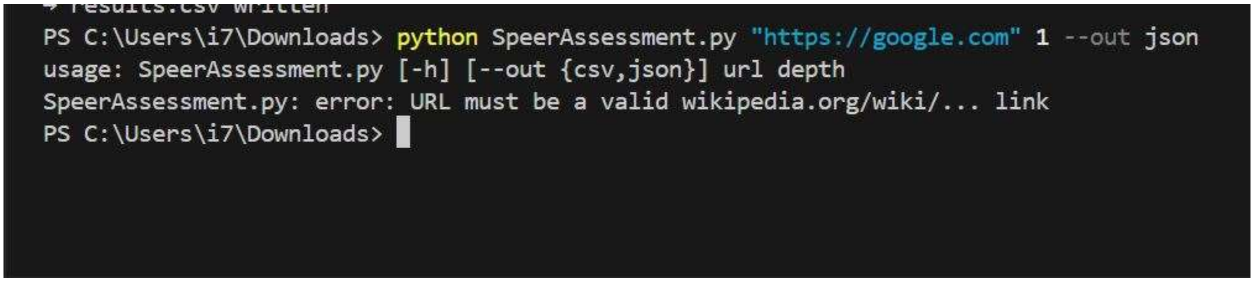
Description:

An attempt was made to start the crawl from a non-Wikipedia URL (`https://google.com`). The script's input validation mechanism is expected to identify this as an invalid link for the specified task.

Expected Output:

The script should immediately terminate and display a clear error message indicating that the provided URL is not a valid Wikipedia link (e.g., "*URL must be a valid wikipedia.org/wiki/... link"*). This confirms the robust input validation.

Actual Output:



```
→ results.csv written
PS C:\Users\i7\Downloads> python SpeerAssessment.py "https://google.com" 1 --out json
usage: SpeerAssessment.py [-h] [--out {csv,json}] url depth
SpeerAssessment.py: error: URL must be a valid wikipedia.org/wiki/... link
PS C:\Users\i7\Downloads> 
```

4. Error Handling: Invalid Depth Integer

This sample showcases the script's validation for the `n` (depth) parameter.

Command Executed:

```
python SpeerAssessment.py "https://en.wikipedia.org/wiki/Canada" 5 --out json
```

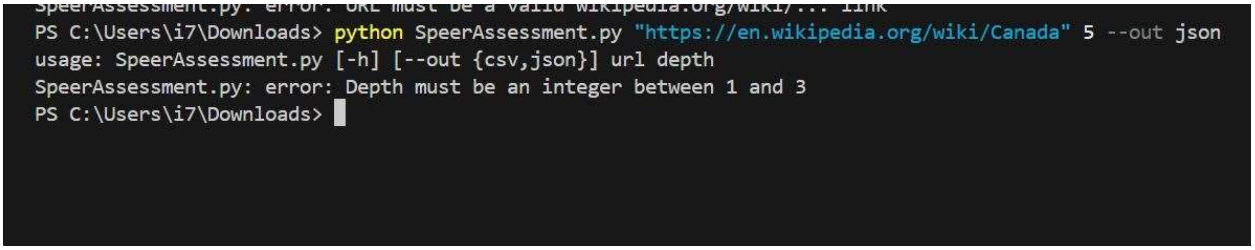
Description:

The script was invoked with an integer `n` (depth) value of 5, which falls outside the acceptable range of 1 to 3 as per the requirements.

Expected Output:

The script should display an error message clearly stating that the depth must be an integer between 1 and 3, and then terminate. This confirms that the script correctly validates the input constraints for `n`.

Actual Output:



```
SpeerAssessment.py: error: URL must be a valid wikipedia.org/wiki/... link
PS C:\Users\i7\Downloads> python SpeerAssessment.py "https://en.wikipedia.org/wiki/Canada" 5 --out json
usage: SpeerAssessment.py [-h] [--out {csv,json}] url depth
SpeerAssessment.py: error: Depth must be an integer between 1 and 3
PS C:\Users\i7\Downloads> 
```