Documentation

Author: Shefali Sharma sharma92@buffalo.edu

Abstract

‘Get Top Topics’ application takes a URL as an input and returns the top topics that have been mentioned in the page to the user. Output is displayed as a list of words.

Application Design & Architecture

Working of Application

1. Client Side- User Interface
   Home page for the web application has been created using HTML, CSS, JQuery and Bootstrap.
2. Server Side - Python Flask has been used as an active server to host the HTML pages. Through Flask, the application takes in the URL as the input from the user.
3. Using NLTK, application parses through the web-page and creates unigrams and bigrams for the words, and using the count of the number of unigrams and bigrams, picks the most significant words.
4. This list of most significant words is then passed to Flask ‘route’, which then passes it to JQuery.
5. JQuery further populates the list on HTML page with the list of significant words, which are then visible to the user.

Project Structure

1. **templates**: Contains all the HTML files:
   - 1.1. layout.html: Base layout which in common in all the pages across the web-application. It also contains links to Bootstrap, jQuery and CSS styles for various HTML tags.
   - 1.2. home.html: Main body which contains the ‘Search bar’ and displays the output in the form of a list to the User.
   - 1.3. about.html: It provides information about the application, as well as the information about the author.
   - 1.4. contact.html: It contains contact information and other links to reach author.
   - 1.5. **includes**: this folder contains features which are directly incorporated in our application. Like:
     - 1.5.1. _navbar.html: Navigation bar to switch between different application pages.
2. **core**: Contains all python source files.

2.1. Processing.py: Class which takes URL as input from app.py. Parses the data on app.py and return a list of topics with the highest word-count.
storeURL.txt temporarily stores the URL given by user for processing by Processing.py.

2.2. Logger.py: Class to write logs for errors/exceptions caught during run-time.
All the logs are stored in '***app.log***'.

2.3. Singleton.py: This is used in Logger.py, so that only one instance for Logger.py could be created. This ensures that for all clients and classes, logs are stored only in a single app.log file.

Setup

Copy the files and folders to your system.

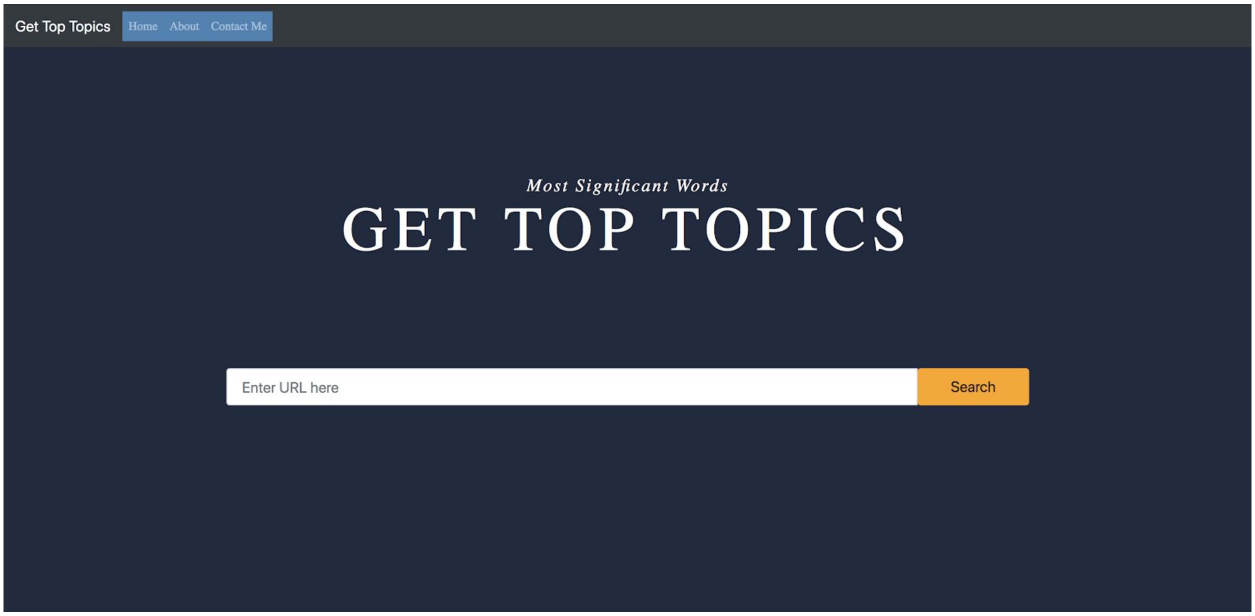
Dependency Management

Navigate to the directory where the files and folders have been copied and install the required dependencies using:

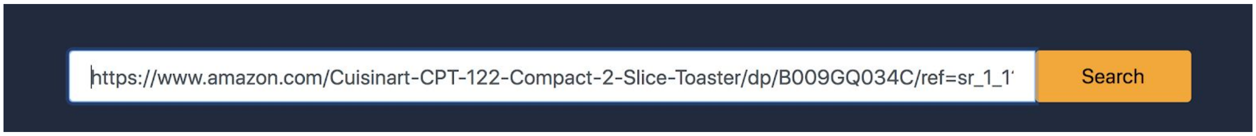
```
pip3 install -r requirements.txt
```

Running Application Server

- Go to the directory where the files and folders have been copied and run command:
  `python app.py`
- Once the application and the debugger have started, look for 'Running on' message on the terminal:
  `* Running on http://127.0.0.1:5000/`
  Take this URL (http://127.0.0.1:5000/) and open it on a browser. You should see the Homepage of the application.

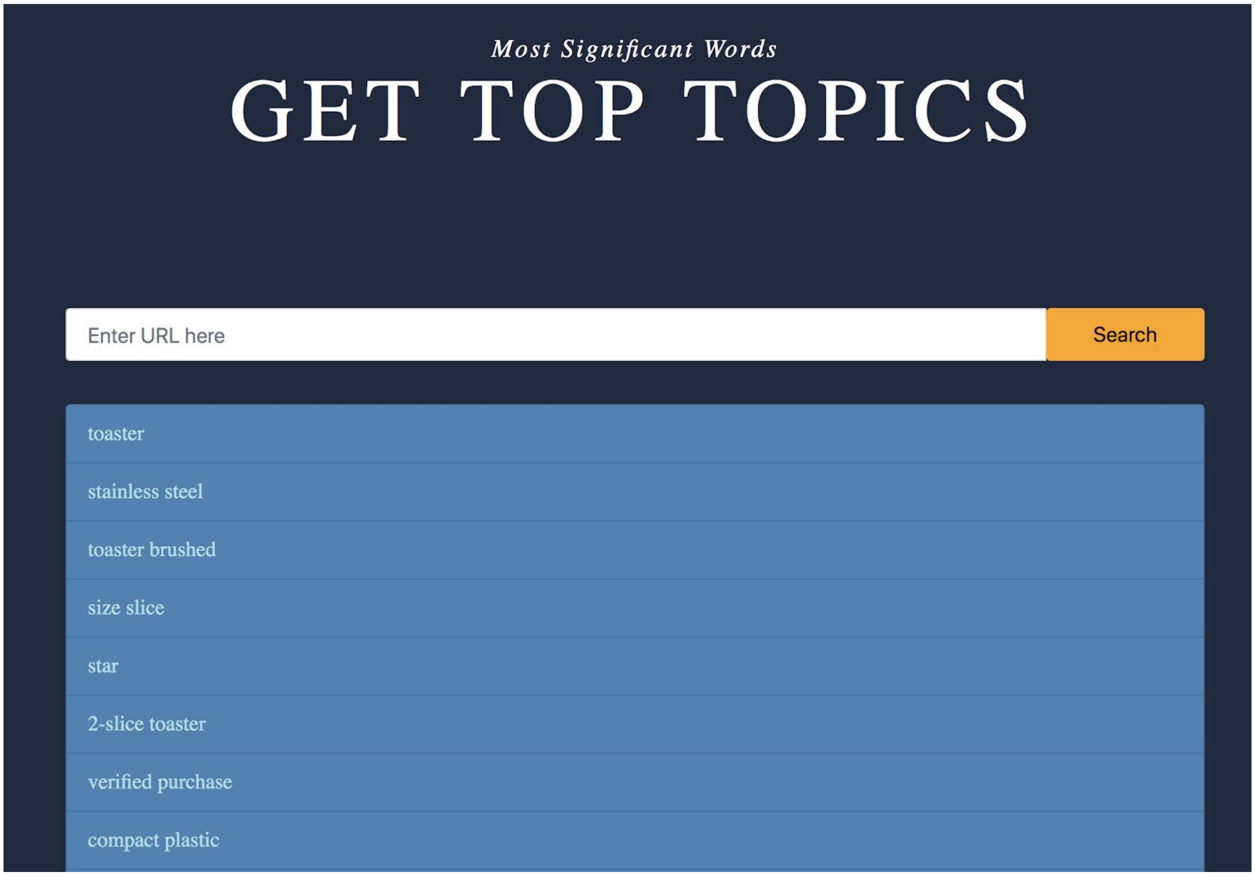


- ## Output Mechanism
- Enter a URL in the input field



https://www.amazon.com/Cuisinart-CPT-122-Compact-2-Slice-Toaster/dp/B009GQ034C/ref=sr_1_1

- When you click on the ‘Search’ button, the topics are displayed below in a list, and input space is emptied for the next URL.



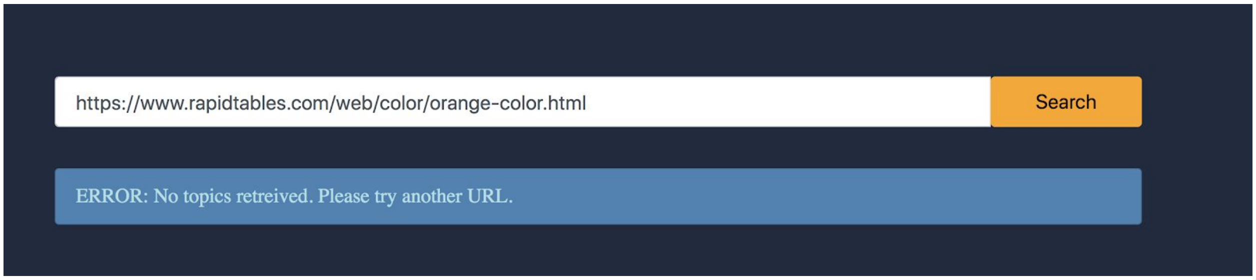
Most Significant Words

GET TOP TOPICS

Enter URL here

toaster
stainless steel
toaster brushed
size slice
star
2-slice toaster
verified purchase
compact plastic

- If you enter an invalid URL , an error message indicating the same will be displayed. format
- If the URL format is correct, but the engine is unable to fetch any words due to invalid URL or network issues or other error, it displays the message saying that it was unable to fetch any topics.



https://www.rapidtables.com/web/color/orange-color.html

ERROR: No topics retreived. Please try another URL.

Logger

Application logs which are generated will be stored in `app.log`.

References:

1. Navigation Bar: Bootstrap ‘Starter Template’ - https://getbootstrap.com/docs/4.1/examples/starter-template/
2. Search bar and list styles: https://www.bootstrapcdn.com/
3. Python Flask: http://flask.pocoo.org/docs/1.0/