

Website Analyzer

02.11.2020

Achyut Tripathi

[MailTo](#)

[LinkedIn](#)

[Github](#)

Overview

A website analyzer is a web or desktop app that samples top level pages of a given website and gathers useful information.

Goals

1. Given a URL, find the homepage of the website (sometimes you may be given a URL of an inner page)
2. Find the links in the home page (and remove duplicates)
3. Store the links
4. For each link, get the web page, extract text and store it
5. Parse the text (remove stop words, punctuation) and generate a list of uni-grams, and bi-grams from the text. We will refer to these as key terms.

Desired Output

Generate a text file containing the following information.

1. The number of top level pages ((a top level page is one that has a link on the home page)
2. Page titles of all the top level pages
3. Meta tags from all top level pages
4. All internal links (links that point to some page in the same website)
5. All external links (links that point to pages outside the site).
6. A frequency table of the top 20 key terms (unigrams and bigrams) in the descending order of frequency and write out a CSV file.
7. Display min, max, average size of each page (including the image sizes, if any)

Problems Encountered

I. How to deal to the webpages which doesn't allow mining

There are lots of webpages present on the internet which don't allow web mining by simply requesting the webpage url by request module. Those

websites throw https 403 forbidden error and for the first time it was a little bit difficult to understand.

To handle this scenario I had to include header which bypass the webpage authorisation code and allow access to the website.

II. How to handle image page urls

During the test run of <https://www.exeterpremedia.com/>, I came to know that there are lots of image urls present in the homepage. On mining those image urls I got unnecessary output. So I researched and tried to figure out all the commonly used image website urls possible to try to neglect those website from mining. Because in this project my main concern was to mine text related data present in the webpage.

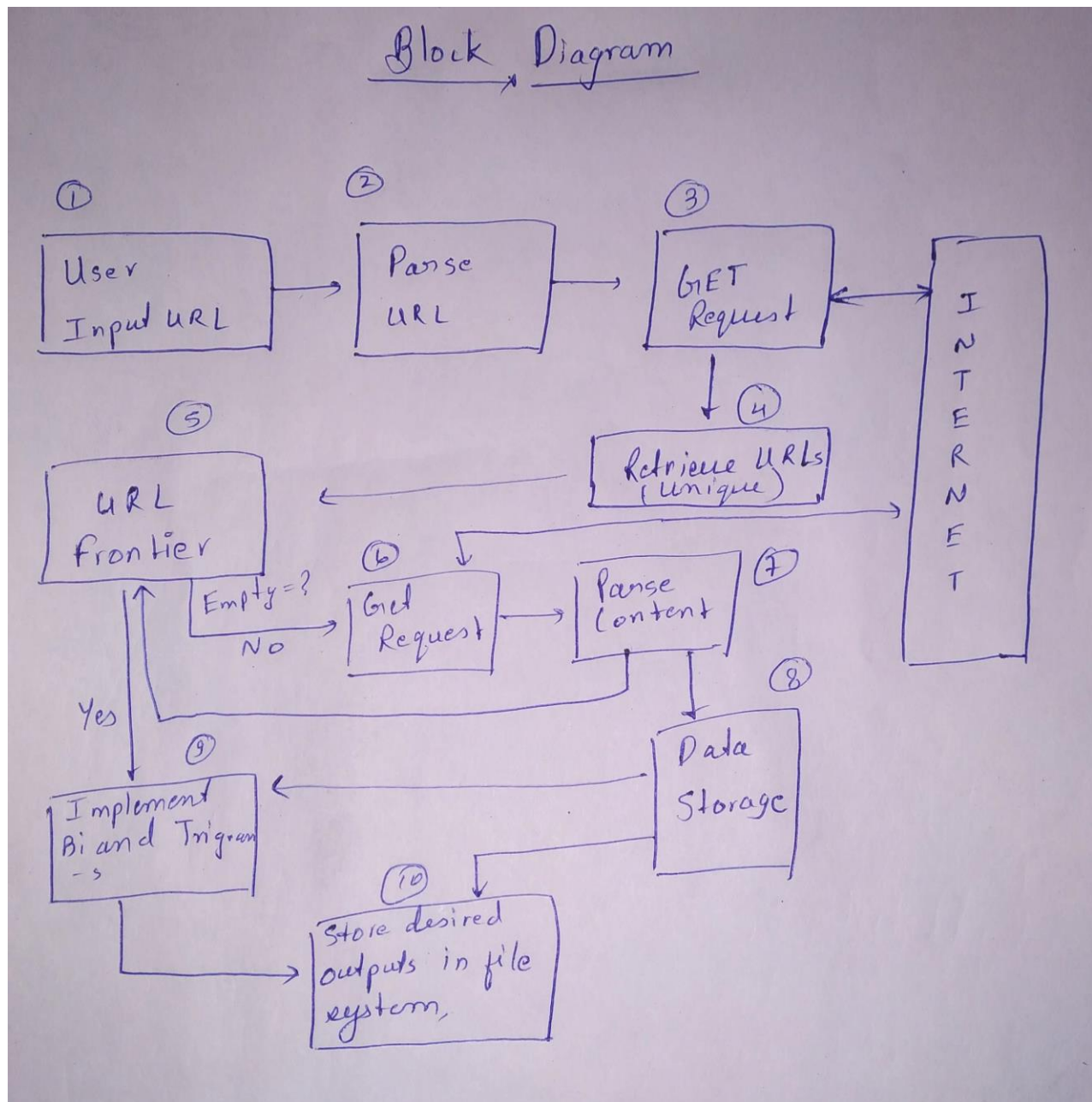
III. How to handle unnecessary content like stopwords

As I progressed in the project, I stored all the content present in the top level websites and generated the bi-grams and tri-grams out of it. After the analysis of top bigrams and trigrams, I came to know that top n grams are words like (is, the) which are the most common words present in a website. So during the content analysis of the website words like is, a, an, the, etc are not very useful. These words are called stopwords. So I tried to remove the possible stopwords from the content by using nltk stopword removal package.

IV. How to store dataframe into the file

One desired output is to store the metadata of each webpage into the text file. As I stored the metadata into the dataframe format, it was quite difficult to store the dataframe into a text file. A simple numerical dataframe can be stored into the text file by converting it into numpy array and then storing it but the metadata is the collection of string data. So for storing the metadata, I stored it into a csv file instead of text file.

Block Diagram



Idioms learned/use

I learned a lot of python associated idioms in the development of this project. I came from the Java development field and switching to python is a whole new adventure for me. There are lots of syntax related changes in python as compared to java or c.

- In this project I learned how to write three lines of code into a single line in python compound statements which can handle if, for and assignment operation into a single line.
- I familiarized myself with anonymous functions that is Lambda function, and how to use them in writing python code.
- I learned the usage of data containers like list, tuple, dictionary and sets which comes very handy in the development.
- I also learned the difference in syntax of file handling from java to python.

Why did you make a certain decision? What were the options?

During the development of this project, there were lots of decisions to make. Whether to use a particular package or not. For example,

- To generate bi-grams and tri-grams of the data we can use nltk packages to do so, but I hard coded the functions to handle each scenario without the use of existing modules.
- Another decision was to pick a perfect code editor of the development. I used Jupyter notebook for this purpose instead of python script file.
- One tough decision was how to store the metadata of each website. After the analysis I saw that apart from the standard attribute list of meta tag, webpages also contained more attributes. So I only stored name, content, http-equiv and charset attributes in the dataframe.
- And the last big decision was how to store the metadata dataframe into the text file. And for that purpose stored the data into csv file instead of text file for more convenient use.

What did you learn

In this website analyzer project I learned a lot of new development techniques and concepts related to python, natural language processing and web mining.

1. I learned how to write a beautiful semantically correct code with the help of flake8 linter.
2. I learned how to fetch webpage content and parse them using BeautifulSoup module.
3. I learned to do basic tasks using nltk package like stopwords removal and word tokenization.
4. I learned and revised my python fundamentals like working with dictionaries, lists, sets and lambda expressions.
5. I learned various concepts of web mining, regular expressions and use of functions and modules for better development of code.

Project Dependencies


Python version 3.7.6 is used in the development of this project.

Modules Required:

- urllib
- re
- requests
- bs4
- pandas
- nltk

Summary of Code Challenge Experience

Exeter Weekend Programming Challenge was one of my best experience in the development field. I learned a lot of great stuff from various concepts of web mining, natural language processing and python. It was a total fun and great learning experience altogether with a steep learning curve. I learned how to code systematically and efficiently within the time frame. Implementing a general web



crawler which can work on any website is already challenging and somehow I did it under the accurate guidance of Ravi and Dorai Sir.

I was totally overwhelmed by the complexity and amount of knowledge this challenge introduced me and made me to grasp that knowledge and work on it to develop the project more efficiently.

It was a great learning experience to the new concepts on applying into a real world scenario.