WIDE

**Topic:** Website independent data extractor.

**Name:** Ayush shah

Prem modi

Vaishvik Sanghavi

Objective: To create a dataset .And using web scraping for automating creation of dataset.

There are two ways of creating dataset from webpage

1) Making use of html response

2) Converting the html response to json format and then playing with keys and values

As of 4^th June we did the 1st method that is, Making use of html response.

urllib is a Python modules that do URL request related stuff but offer different functionalities.

1. urllib provides the urlencode method which is used for the generation of GET query strings, urllib2 doesn't have such a function. This is one of the reasons why urllib is often used along with urllib2.

Requests - Requests’ is a simple, easy-to-use HTTP library written in Python.

2) Python Requests encodes the parameters automatically so you just pass them as simple arguments, unlike in the case of urllib, where you need to use the method urllib.encode() to encode the parameters before passing them.

3) It automatically decoded the response into unicode

4) Requests also has far more convenient error handling .If your authentication failed, urllib2 would raise a urllib2.URLError, while Requests would return a normal response object, as expected. All you have to see if the request was successful by boolean response.ok

Thus we used requests rather than urlib

As of 4th june we did the first method

well the answer is you should try to parse couple of pages with HtmlSelector by Using beautiful Soup. and find some stats.

2ndly most of people use beautiful Soup for parsing

Scrapy basic motive is Crawling hence we can go with beautiful Soup.

Scrapy is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [web-crawling](https://en.wikipedia.org/wiki/Web_crawler) [framework](https://en.wikipedia.org/wiki/Web_framework) written in Python. Originally designed for web scraping, it can also be used to extract data using [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) or as a general-purpose web crawler. It is currently maintained by [Scrapinghub Ltd.](https://en.wikipedia.org/w/index.php?title=Scrapinghub&action=edit&redlink=1), a web-scraping development and services company.

Scrapy project architecture is built around "spiders", which are self-contained crawlers that are given a set of instructions.

**Beautiful Soup** is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language))package for parsing [HTML](https://en.wikipedia.org/wiki/HTML)and [XML](https://en.wikipedia.org/wiki/XML) documents (including having malformed markup, i.e. non-closed tags, so named after [tag soup](https://en.wikipedia.org/wiki/Tag_soup)). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for [web scraping](https://en.wikipedia.org/wiki/Web_scraping).[[1]](https://en.wikipedia.org/wiki/Beautiful_Soup_(HTML_parser)" \l "cite_note-crummy.com-1)

It is available for Python 2.7 and Python 3.

Short overview of the idea,

Data has been administered by Appling various filters to get the data of our choice.

Until 14th july we made a web craper which extracts the relevant information from the web page to excel sheet without using json.

We customized our website to garner various types from the websites like

1. Personal Websites

2. Photo Sharing Websites

3. Writers / Authors Websites

4. Community Building Websites

5. Mobile Device Websites

6. Blogs

7. Informational Websites

8. Online Business Brochure/Catalog

9. E-commerce Websites

Etc.

The software application was tested on various websites like

Youtube ,Myntra ,Twitter ,Blogspot , Jabong ,LinkedIn , Newpapers like the Hindu,Dictionaries,Flick ,Lenskart ,Etc

And we were successful in achieving high accuracy on these websites.

Few Problems Faced:-

Limited server request and limited website access by not using heavy load libraries and using simple yet robust libraries.

No defined ways to extract information and also the independency of the website created a major issue.

We solve the key value iteration that we might face by using html response alternative

 The project helps in extraction of data from website based on the inputted URL

It helps to filter the important data and provides a specific data set.

Lack of data set is solved by this work. It automates the task of creating the dataset .It can be Can be used by any person to create a dataset of the website he wants to

We have applied various filters to get the data of our choice and relevance .We also have got the most relevant data by sorting with frequency of that word occuring.

Web Scraping and Limiting Server Requests:

Believe it or not, some people that work in the IT department of the website that you are trying to scrape do not want you to hammer their servers with requests. Every time you request a page, the server needs to respond with a web page. If you are making thousands of requests a minute, the server is not going to be happy. Therefore, most servers will deny access to your IP address if you continue to inundate them with requests. This denial of access can be temporary or you can get a total ban. This is one of the web scraping problems that can considerably slow down you web data extraction.

limiting of server requests for web scrapers:-

First of all, the web scraper should not be hammering a server so hard in the first place. It is nice to be polite, so no hammering. Web scrapers can reduce the number of requests sent in a minute. We should set it to a reasonable amount as not to flood the server with requests. If you do need to hammer a website for whatever reason, you can do so from different IP addresses. There are proxy services in which you can subscribe to firstly hide your own IP address and secondly use a multitude of different IP addresses. It is possible that the target website has already blacklisted these proxies, but you will know soon enough.

Web Scraping and Limited Website Access

This is a new web scraping problem for Entropy Web Scraping. Some websites, even though you have a user login and password and you have no problem authenticating, will not allow access to certain parts of the website. For example, consider LinkedIn. Sure, you can set up a fake account and log in. The problem is, in order to view anyone’s profile, you need to develop your own network. This means connecting to other users and building up a network. It is unlikely that other users on the social network are going to accept an invitation from spammy looking fake accounts.

Limited website access for web scrapers:-

Well, without putting in the effort to create a pretty good looking professional profile and adding people over time, there is not much that can be done in this situation. Sure, the web scraper could use their personal profile to complete web scraping, but there is a risk of getting banned and losing their anonymity. This web scraping headache is one yet to be overcome.

Web Scraping and Poorly Constructed HTML

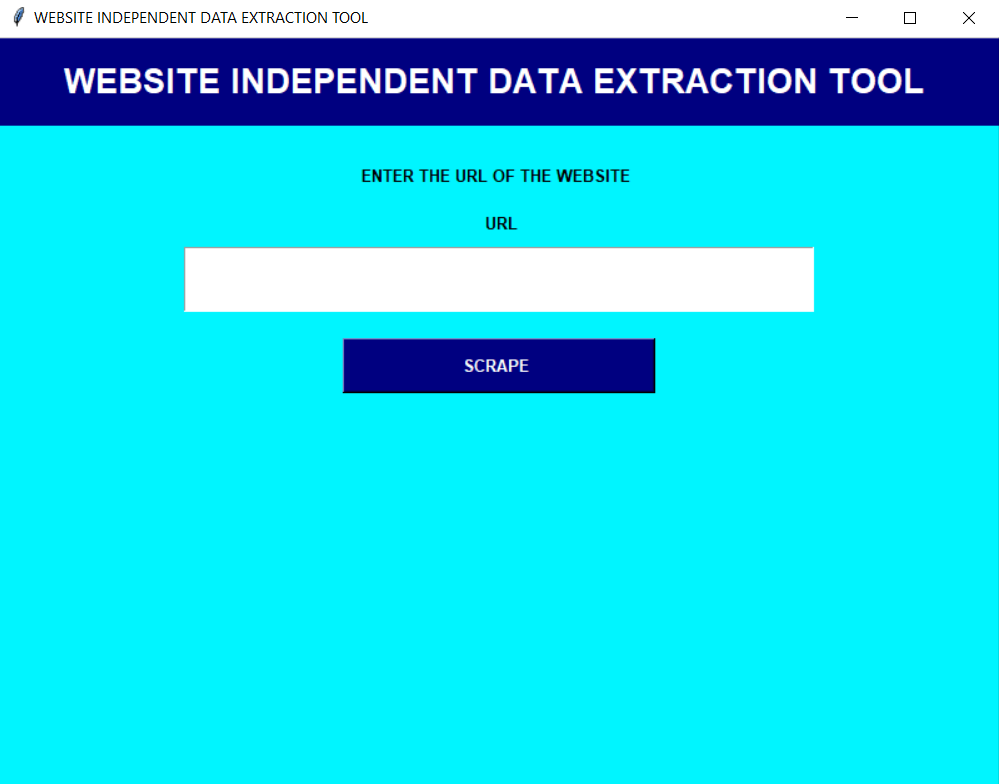
This one of the web scraping problems that is new for Entropy Web Scraping. Recently, when web scraping a government website, we were horrified to find how poorly the website had been built. Usually, websites are created with sophisticated frameworks or backend technologies that do things like construct a web page on the server side and then render it to the user. Recently, we have discovered that this is not always the case. It seems someone in the Australian government has been creating websites by hand… That’s right, manually writing HTML elements such as tables, table rows, table elements etc. Not only would have this been very time consuming for the person who did this mind numbing task, but it is also a headache for the web scraper!

Poorly constructed HTML web scrapers:-

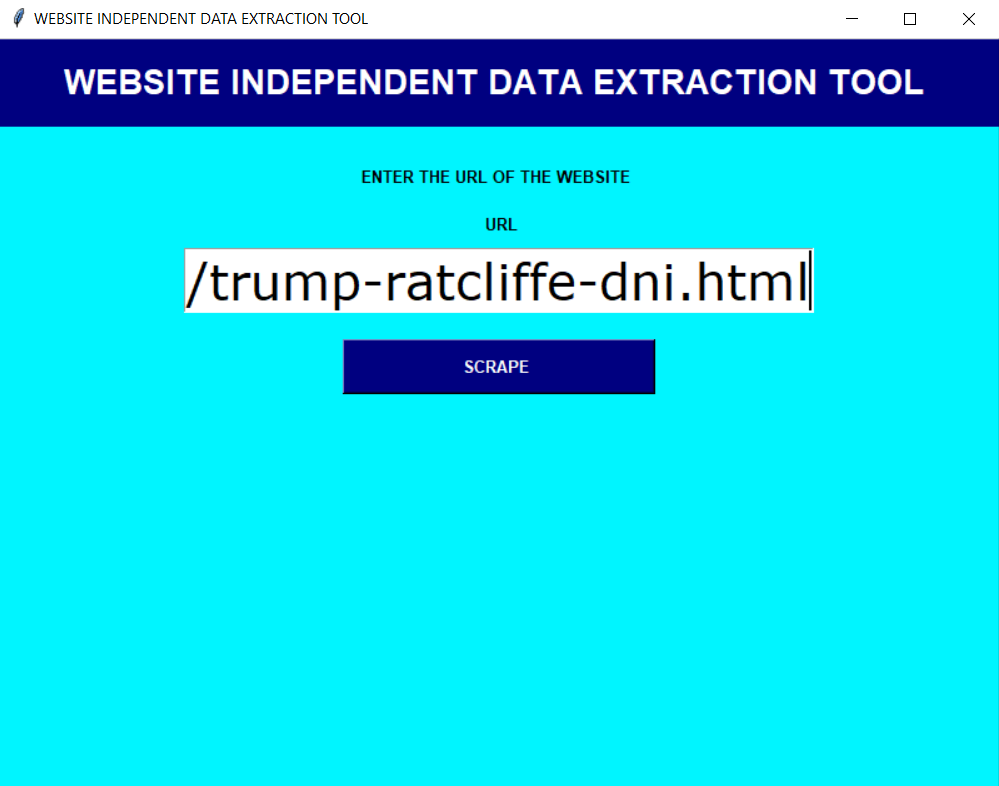
This means that it is hard to use our document parsing technologies to extract content from the document. XPATH and CSS selectors do not work for a document that does not consistently follow any pattern or structure. In the example above, we had to resort to extracting the entire text of the page and use regular expressions (text pattern matching) to extract the relevant data. For what seemed like a very easy web scraping project at the start ended up being very frustrating and time-consuming.

**MANUAL:**

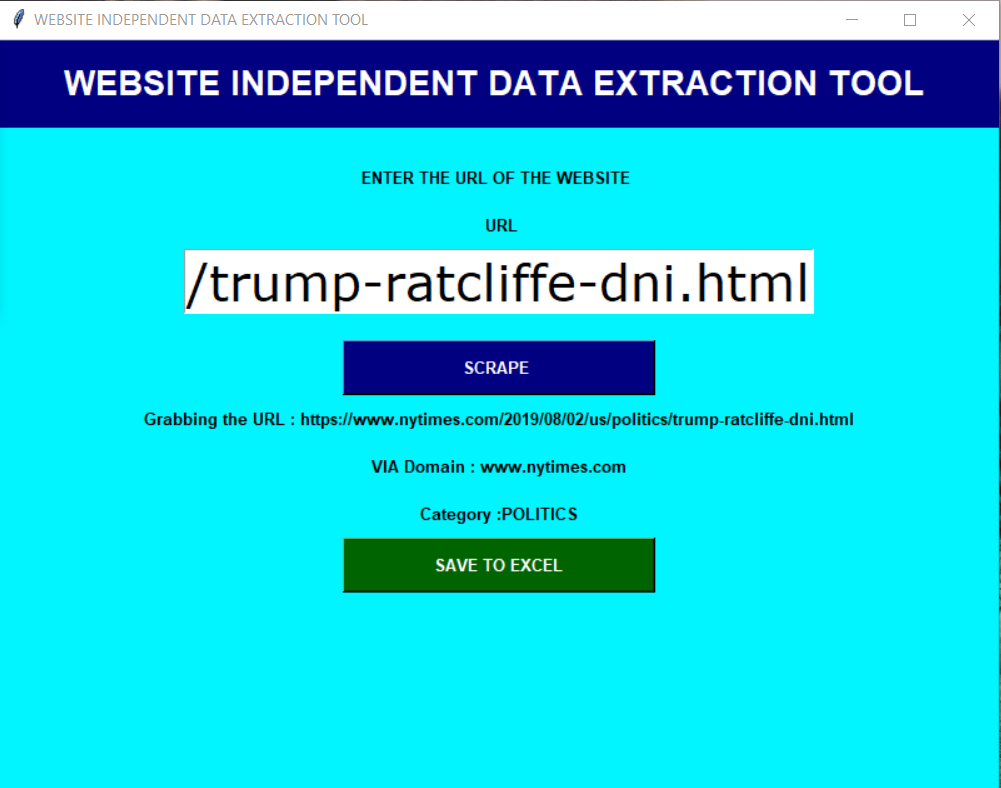
Step 1.) Start the software



Step 2) Copy and paste the link of the web-site you want.

****

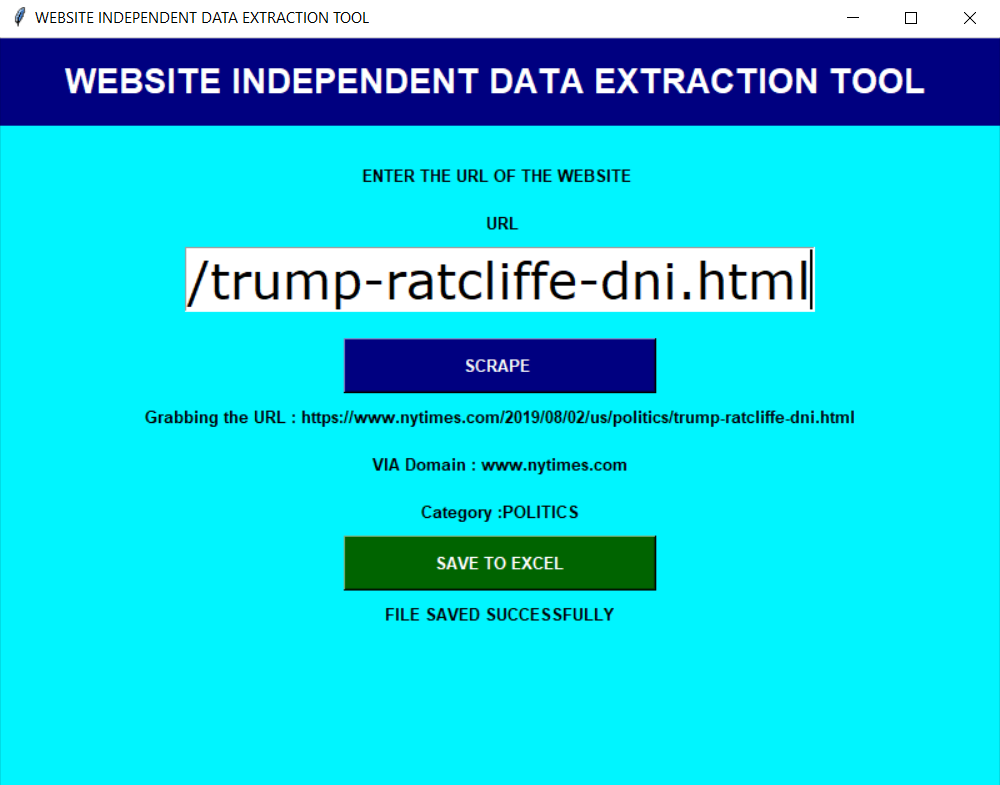
Step 3) Click on the SCRAPE button

****

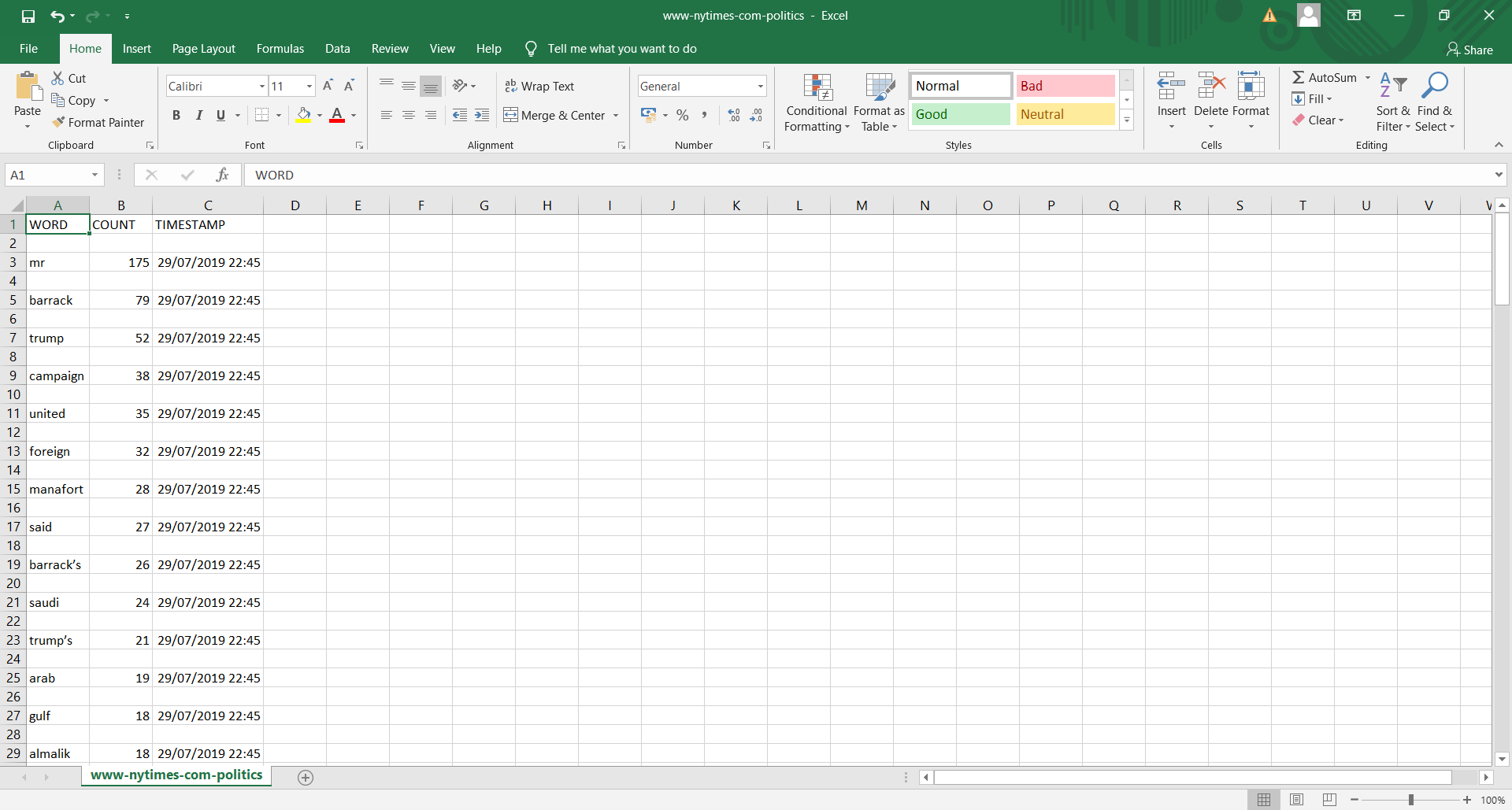
Step 4) This Output format will be generated.

****

Step 5) Click on the SAVE TO EXCEL button for generating an excel file containing output.

****

Step 6) Such an Excel file will be saved.

****

WORKING/ARCHITECTURE:

Start

Get html response

Get body of html

Process the text to get

keywords

Apply filters on keywords

End

Get the top 30 words

based on frequency

Get frequency count of

filters

Display the result

to the user

Close the GUI

**THANK YOU**