# Webscrape.py

The webscrape.py file does the following:

1. Hierarchically crawls the 211sonoma.org page looking for all links not in this domain. Only record links that aren’t blank.
2. Records all words in the hierarchy as it traversed the tree to find the link. This means all words found on the “way down” are saved and applied to all external links on a page.
3. Split any words found that contain an “/”
4. Remove all duplicate words
5. Remove common words that won’t help with search for meaning:  
    {"About","Us","Nosotros","para","Sonoma","County","in","de","en","la","un","una","su","sus","y","&","with","[www.211sonoma.org](http://www.211sonoma.org/)","of","the","2-1-1","and","for","of","para","Click","here","","http:","www.211.org","-","del","el","sin","al","con","los","las","a","El","por","by","e","or","at","o","to","it","A","as"}
6. Remove duplicate links.
7. Write out “Words.sql” SQL file with all unique words and an index.
8. Write out “Links.sql” SQL file with all links and their title
9. Write out “Wordlinks.sql” SQL file with all words referenced for each link.

These SQL files, when in an SQL database allow you to find links for various input words.

The following query is an example of how to grab links base on key words. (This example looked for all links which had the word “Lunch” referenced.)

SELECT links.links AS links, links.title AS title  
FROM `key\_words`, `links`, `word\_links`  
WHERE key\_words.words = "Lunch"   
AND key\_words.ID = word\_links.word  
AND word\_links.link = links.ID

And here are the example results:

## Dumping data for table links

|  |  |
| --- | --- |
| https://www.google.com/maps/d/u/0/viewer?mid=12sGarVKPhpZMJsoQNhED65mJilzG6245&ll=38.51774044915148%2C-122.9129044&z=10 | SCOE School Lunch Sites (COVID-19) - Google My Maps |
| https://sonomacounty.ca.gov/Health/Information-About-Coronavirus/ | Sucuri WebSite Firewall - Access Denied |