Coursework 2 Web Services and Web Data

Web crawling tool

Georgios Michailidis sc18gm@leeds.ac.uk

Crawling the website:

To crawl the website, it means that you access each section of the website and retrieve the information contained in each section. In order to collect the information, you must access the links, thus the collection of links containing the relevant information for us to crawl needed to be obtained. To obtain the links we needed to connect to the given website and then extract the links. That was done with help of 'Requests' library, also the 'Beautiful Soup' library was used to obtain the links contained in the html source code of the website. The collection of the links that contain the page index was needed since each page contains the links of the website which those links provide the information regarding the website which will help build the inverted index. Thus, iterating through the pages and receiving all the links contained in each page. The corresponding links were saved in a list which was filtered to remove any duplicates in case of receiving a link twice which that will have led to accessing the same link twice thus receiving the same information twice. Two files were created 1 has all the links of the website (unique, all_link_List) and the other one (Link_List) has the information (countries and continents URLs) which we are going to use to create our inverted index. Also, a 5 second politeness window was placed between successive requests to the website.

Inverted index:

An inverted index is a database index storing a mapping from content such as words or phrases to its location in a document or links in our case. To be more precise an inverted index in this case is the mapping of the words from where are located within a link which corresponds to a location on the website. Inverted index is used for fast search since each word has a list that corresponds to where the word is located and how many times it appears thus it does not need to go through a document to locate it which this can be time-wasting if the data sets are too large.

To create an inverted index, we needed to obtain the relevant information from the website and then filtered it out. Analytically, symbols, stop words, punctuations needed to be removed since they do not provide any valuable information and it is most likely that people will not try to search for those words or symbols. Also, numbers were not needed, and they needed to be removed. The information that the inverted index holds is regarding the countries thus filtering out the irrelevant links was needed. The result was 253 links, 246 links which refer to each country and 7 links for the

continents these links were saved in a file. After crawling the corresponding links, the information was received from the website as text. Then filtered out the symbols, stop words, punctuations and tokenize the information of each country into a list as words. Afterwards, looping through the list and recording how many times each word appears and at which corresponding link it appears. Furthermore, each word has its corresponding inverted index and the frequency which appears in the location (location=links, score=frequency) as a dictionary. Finally, the inverted index is appended in a file with each word placed in a different line with the corresponding inverted index which includes a list of how many times the word appeared in the corresponding link. If it appeared in a different link, then the new link is recorded to the list with the frequency of the word. For example, United [3:1, 4:1, 5:1] which means the word appeared in link 3 one time in link 4 one time and in link 5 one time also. As I have mentioned before links used for the inverted index exist in Link_List file and are all sorted in ascending order, for better understanding of the user an inspection of the link file might be helpful.

Computing the scores of pages:

The presentation of the links regarding the words placed in the find function works as follow. Firstly, the user imports the word or words of his/her preference and then the links appear regarding the selected words. For combination of words only the common links appears. The choices of the user are collected and then saved into a list. Afterwards the comparison from a list of links takes place and if the word is located then the corresponding links appear. If 2 words are placed, then a comparison between the two words take place to find a link that both appear in it.

Usage of the search tool:

The tool has 4 functions, build, load, print and find. The tool has a welcoming interface asking the user to choose what functions he/she would like to access. However, if the build function was not firstly initiate to build the inverted index, then the rest of the functions will not be applicable. The reason is that load, loads the inverted index in the terminal from the file printing each word with the corresponding inverted index. The print function prints the word which the user entered and its corresponding index. Finally, the find function where the user places a word of his/her choice and then a list of the corresponding links appears with those that have the most frequency of the word placed at the top. However, if the word appears the same number of times in different links that means that the order of the links does not matter. After the execution of each function the tool terminates except if you have placed a wrong input in that case you are re-directed to the function. Also, the selection of function is case sensitive thus it is needed to be written as seen in the welcoming interface.