**SEO Analyzer**

**Goal:**

To create a SEO analyser which takes a web URL as an input and returns the common keywords based on the topic from the given URL.

**Functionality:**

* Takes a URL as input and returns the 3 word phrases and 4 words phrases as keywords based on the content of the page of the URL.
* Parse the web URL, if its valid then get the raw HTML data and analyse it further for the common keywords based on the content.

**Dependencies:**

I have used an external java library – jSoup for parsing the URL and getting the HTML content from the URL.

**Design and Implementation:**

The important elements in a webpage can be identified through <Meta Description=””>, <Title>, <h1>, <Img alt=””>, url path and the <body> content. The data present inside these tags are generally considered as the main elements of the page and almost every webpage will have content inside these tags. So, when I get an URL as an input, I ping the URL using the jSoup library to check whether it is a valid URL or not. If it’s not a valid URL then the program exits throwing the exception error message. If it’s a valid URL, then it parses the raw HTML contents and collects information from the above-mentioned tags and stores in a string.

Then, the string is converted into 2, 3 and 4 word phrases and stored in a Hashmap using recursion which the phrases as its key and number of occurrence as its value. And the words which are less than 2 characters are ignored. And the program is structured in a way that it can be modified if you just want 3 letter keywords or 4 letter keywords based on your preference. For now, it generates 2, 3 and 4 word phrases and later it can be modified by getting those inputs as arguments.

Once all the phrases as stored in the Hashmap, we can get the commonly used phrases based on the number of occurrences. I have considered that the first 1000 words of the body in a page will have the important and most relevant data related to that topic. This also helps to remove all the clutters from the page. So, when the body of the page has lesser than 1000 words then the phrases which occur more than thrice are considered as main keywords/ phrases. If it has more than 1000 words, then the number of occurrences must be 5 to be considered as main keywords/ phrases.

**How to run:**

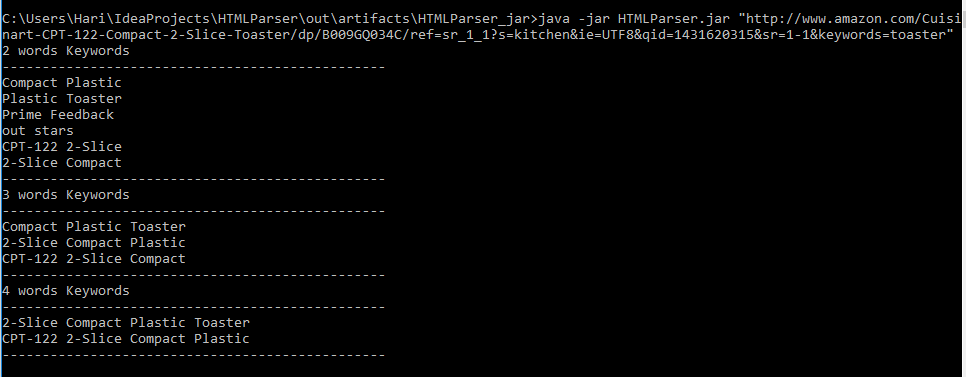
I have attached the .jar file along with this document. It can be run from the command line using the following command

*Java -jar HTMLParser.jar <URL>*

For example,

java -jar HTMLParser.jar “<http://www.amazon.com/Cuisinart-CPT-122-Compact-2-Slice-Toaster/dp/B009GQ034C/ref=sr_1_1?s=kitchen&ie=UTF8&qid=1431620315&sr=1-1&keywords=toaster>”

**Sample output:**

****

**Ways to improve:**

We can have a set of prepositions and conjunctions that can be ignored from the phrases. This can be done having these connectors in a hashset and before adding the phrases to the hashmap, we can ignore it if it has those connectors.