**Information Retrieval – Homework1 (Build a web crawler)**

**1) Briefly compare the results obtained from the two crawls in terms of URL overlap, perceived quality and efficiency aspect, coverage of the crawl topic (i.e., solar eclipse).**

Crawl – BFS

The BFS crawl explores nodes at each depth before moving onto the next depth level. More subtly it regards the nodes in the shallow level to be more import than the nodes at the deeper levels. BFS search run with the seed URL - <https://en.wikipedia.org/wiki/Solar_eclipse> returns at most 1000 unique links. In terms of the accuracy or the relevance of content, BFS algorithm crawls we pages that return more valuable information related to the seed. The content in the seed page is closely related to the seed itself. Since BFS explore the entire depth before moving on to the next depth, maximum relevance is almost achieved in one depth. On the downside the information retrieved by the crawl could be vague or unrelated to the seed as the depth deepens. The content becomes less related to the seed URL but becomes more related to their immediate parent URLS. In terms of speed, there is a heavy dependency on the structure and the goal. Since the goal here is not specific, and we do not know how closer is the goal to the seed, BFS and DFS fairly perform the same (There could be cases where 1000 links are reached sooner in BFS than in DFS).

Crawl – DFS

The DFS crawl explores nodes along a branch before backtracking to the former depth. More subtly it regards the nodes in the deeper level to be more important than the ones that are shallow. DFS search run with the seed URL - <https://en.wikipedia.org/wiki/Solar_eclipse> returns at most 1000 unique links. In terms of accuracy or relevance of content, DFS algorithm can find specific data from one page to another in relatively few hops. Since DFS explores the entire branch before backtracking, we can argue that the content collected by BFS could arguably be more related to the seed URL than DFS. But there could be cases where the goal is reached faster in DFS and hence crawling relevant content than BFS. In terms of speed, the usage and the number of links present in the document might slow or speed up the process of DFS.

In a broader sense, the efficiency, the speed of the algorithm and the quality of the content depends on the implementation, data structures used, the data that we are targeting and the output expected.

**2) Focused Crawling**

Key word variation process:

* Collected all the words after the “/wiki/” pattern by striping the string (URL) and the anchor text
* For the words collected, special characters were removed (Non-English and numbers) and replaced with a space
* For the given keyword or the set of keywords, checked if the keyword/set of keywords are present in the words collected
* If yes, then the words are split (By removing the keyword from the word), leaving the parts before the keyword and after.
* Using the PyEnchant library, the parts are checked if they are valid English words
  + For example,
  + Reference – Piazza
  + Keyword = “Moon”
  + Words = “fumoon” and “fullmoon”
  + Check if PyEnchant consist of “fumoon” and “fullmoon”
  + “fumoon” will be False and “fullmoon” will be True
  + Split “fumoon” into “fu” and “moon”, check PyEnchant again for both the words
  + Split “fullmoon” into “full” and “moon”, check PyEnchant again for both the words
* If all the above conditions are true, then the words are considered valid (Also, the URL is considered valid to be crawled)
* List of valid URLs are used for the BFS crawl