LT2 | Balagon, Val Anthony V. | Borromeo, Chloe B. | Dorado, Joshua Renzo R. | Tallo, Francis M.

**DMW Lab Report 1 - Supplementary Guide**

**Part I: Basic Goals**

The code we developed is split into two main categories of functions: (1) Main Functions that handle web crawling and result generation, and (2) Secondary Functions that support the Main Functions. The full docstring per function is available in the accompanying Jupyter notebook.

**Table 1. Main Functions**

|  |  |
| --- | --- |
| **Function & Parameters** | **Description** |
| web\_crawler(a, dir\_, url**=None**, lang**=**'en', linkhist**=**{}) | This function starts from a random Wikipedia page (unless a start url is passed) and follows the first page link until it reaches the Philosophy page, a page with no links, or loops back to a previously visited link. It accepts the chosen language (lang) defaulting to English Wikipedia. It also accepts a Link:DoS (linkhist) dictionary which shorts the path to the target page (Philosophy for English), bypassing redundant crawls from previous iterations (through the crawl\_iter function).  Consistent with the basic goals, certain links are excluded (e.g., italics usually language links, disambiguation pages, external links, etc.) during the crawl. In finding the first valid link, it also prioritizes links within the article’s main body.  Returns  -------  flinks: list  List of visited page URLs during the crawl (including the start page).  dosl: list  List of degrees of separation (DoS) for each visited page.  msg: str  Message indicating the status of the crawl (e.g., 'OK - normal path',  'NOK - no links found', etc.).  Apart from the returns, the crawl result is saved into a CSV file. |
| crawl\_iter(basedir, urls**=**list(), lang**=**'en', a**=**0, b**=**100, test**=False**) | This function iteratively calls the web\_crawler function. It also loads a Link:DoS dictionary which shorts the path to the target page in web\_crawler, avoiding redundant crawls from previous iterations.  Returns  -------  df: pandas.DataFrame  DataFrame containing consolidated crawl results (one crawl per row).  Apart from the returns, the updated Link:DoS dictionary and the consolidated crawl results are consistently backed up into CSV files for future reuse. |

The Main Functions, in particular, crawl\_iter, is used to generate the consolidated crawl results.

**Figure 1. crawl\_iter Sample Cell Output**

A screen shot of a computer

Description automatically generated

The cell output prints the progress for ease of tracking.

**Figure 2. crawl\_iter Sample Cell Output (w/ Short Path)**

A screenshot of a computer program

Description automatically generated

In a crawl, when the current link has been visited from a previous iteration (or from the loaded linkhist), it will immediately break the crawl and generate results for the iteration,

**Figure 3. crawl\_iter Sample Consolidated Crawl Result (CSV)**

A screenshot of a phone

Description automatically generated

The consolidated crawl result (CSV) contains the crawl results per row (iteration), with 4 columns – Start (start page), DoS, Path (visited links from start page), and Msg (details of the crawl).

**Table 2. Secondary Functions**

|  |  |
| --- | --- |
| **Function** | **Description** |
| get\_parent(tag, pfr\_list**=True**): | It iteratively retrieves the parent tag of a given HTML tag until it reaches ['p', 'li', 'th', 'td'] (prioritized for finding valid links) if pfr\_list is True. Otherwise, it just returns the immediate parent. If the tag 'i' is found in the parent-finding, it is immediately returned (since italics are not valid links).  Returns  -------  parent: bs4.element.Tag or None  A parent tag given the input tag. Returns None if no parent is found. |
| enclosed(tag, pfr\_list**=True**) | It checks if an HTML element is enclosed within parentheses or italicized. It uses the get\_parent function to expand its search to the parent element.  Returns  -------  bool  True if the tag is enclosed within parentheses or italicized, False otherwise. |
| online(alink, url): | It checks if an anchor link is online and accessible.  Returns  -------  bool  True if the link is accessible, False otherwise. |
| wiki(alink, stop\_words) | It checks if an anchor link is a valid Wikipedia link.  Returns  -------  bool  True if the link is a valid Wikipedia link, False otherwise. |
| get\_link(body, stop\_words, url) | Retrieves the first valid link within the primary tags (p, li, th, td) of the main body.  Returns  -------  link: str  The URL of the first valid link found. |
| get\_other\_link(body, extags, stop\_words, url) | Retrieves a valid link outside the primary tags (if get\_link has returned ‘’).  Returns  -------  link: str  The URL of the first valid link found (outside the primary tags). |
| save\_linkhist(linkhist, dir\_, a, iter, fin**=False**) | Saves the Link:DoS history to a CSV file for future use (short paths). |
| save\_conso(data, dir\_, a, iter, fin**=False**) | Saves consolidated crawl results to a CSV file. |

**Part II: Advanced Goals**