**Financial crawler documentation**

This program allows an operator to input a list of keywords, a positive- and negative-weighted wordlist, and a list of URLs to start crawling from. It scans every web page it comes across for the keywords. If a group appears somewhere in the text of the page, then the article is pulled and the text examined for the existence of the positive or negative words in paragraphs that contain at least one of the keywords. Optionally, a date or range of dates can be specified for each group of keywords, which the crawler will match against the date of the article it is analyzing.

**Usage**

1. Three text files must exist in the same folder of crawler.pl. These files are:
   1. positive.txt
   2. negative.txt
   3. websites.txt

A description of the format of these files follow. The names must be all lower-case.

1. An input file listing the keywords to search for must be somewhere on the computer. This file can be named anything and is provided when invoking the program.
2. Execute shell.bat in the folder. A command prompt window should appear.
3. For a basic crawl, use the format:

perl crawler.pl <input file> <output file>

Where <input file> and <output file> represent filenames. The output file will be created if it does not exist. If it does exist, output will be appended to the end of the file. Optional arguments can be passed to the program to modify certain behavior, in the format of a hyphen (-) and a character, or two hyphens (--) and a word. These arguments are detailed below.

1. By default, output only appears when a page is weighted to a non-zero value. A description of this weighting process follows.
2. Quit the program and print a summary of the crawl by pressing CTRL+C.

**Input files**

* Note: due to quirks in formatting with Microsoft Word, the suggest method to edit the input files is with a text editor like Notepad.
* <input file> – The main input file containing groups of keywords to be searched. Each line of this file represents a different keyword group. All keywords in a group must be present in a web page to be considered relevant to the search. Multiple words may be quoted to be considered one word, using double quotes (keep in mind that the typographer’s quotes that Microsoft Word uses will not be read. See this [link](https://office.microsoft.com/en-us/word-help/change-curly-quotes-to-straight-quotes-and-vice-versa-HP005190124.aspx)).

For example, an input file may contain:

Facebook acquisition Instagram

"Google, Inc."

In the preceding example, the crawler will look at all pages which either contain “Facebook,” “acquisition,” and “Instagram” in any order throughout the page, or which contain “Google, Inc.” in exactly that order.

Additionally, a one or two dates may be added to each group of keywords to indicate that the article date is relevant to the search. The date format is in the form of YYYY-MM-DD (for example: 1980-07-01 for July 1st, 1980). If one date is used, it is considered to be the starting date of the search, and only articles from that date onward are used. If two dates are used, then only articles with a date within that range are used.

For example, to make the previous input file consider dates:

Facebook acquisition Instagram 2012-08-14

"Google, Inc." 2004-08-19 2005-08-19

Now, only articles containing the keywords “Facebook,” “acquisition,” and “Instagram” which were published after August 14, 2012 (the date the acquisition occurred) will be searched. For the second set of keywords, only articles containing “Google, Inc.” that were published within a year of its IPO will be considered.

* positive.txt, negative.txt – Each line of these files contains exactly one word and then a numerical weight of that word. Note that the negative words do not require a negative weight—the number is assumed to be negative.

For example, given these two word lists:

|  |  |
| --- | --- |
| positive.txt | negative.txt |
| okay 0.1  good 0.3  exemplary 0.9 | mediocre 0.4  deplorable 0.9 |

When an article is found containing the keywords listed in the input file, each paragraph is scanned for *at least one* keyword and *one or more* positive or negative keywords. The weight of each positive or negative keyword is then summed up to produce a net weight for the page. If this weight is positive, the page is deemed to have a positive sentiment. If the weight is negative, the page is deemed to have a negative sentiment.

If, for example, a paragraph contained a keyword as well as “good” twice, and “deplorable” once, the weight of that paragraph would be calculated as: 0.3 + 0.3 – 0.9 = -0.3. If the page only contained that one paragraph of relevant information, the page weight would be -0.3 and hence would have negative sentiment toward the keyword.

* websites.txt – Each line represents a URL that the crawler will use to start its search. After analyzing the URL for content relevant to the search, it will then follow all the links that are inside the domain. For example, if the initial URL is CNN.com, it will not follow links to NYTimes.com.

When providing links, the “http://www.” is required and no forward slash at the end is necessary. For example, “http://www.cnn.com” will instruct the program to start its search at CNN.com. Multiple URLs can be specified.

**Output files**

During the course of a search, the program will write to two files: <output file>, which is specified when the program is run, and crawler.log, a log file for the program. Timestamps are used to organize the contents of these files over multiple runs. The contents of these files are described below:

* <output file> – This file compiles all relevant statistical information found during a crawl. Specifically, it records when a website that has been deemed relevant to the search has a non-zero weight. The format of this record is:

<KEYWORDS>: <SENTIMENT> (<WEIGHT>) – <DATE> – <URL>

Where:

<KEYWORDS> is the keyword group for which the page is relevant to

<SENTIMENT> is either POSITIVE or NEGATIVE, based on the weight.

<WEIGHT> is a positive or negative weight derived from the sum of relevant positive or negative words, based on the weights those words are given in the respective input files.

<DATE> is the publication date of the page, only displayed if a date or range of dates are specified for the keyword group.

<URL> is a link to the actual web page for reference.

In addition to these records, a summary is printed after the crawl, which lists the overall results for each keyword. For example, for the following summary:

Google

+3 -- 4 pages

-1 -- 2 pages

Skipped: 1

The keyword group Google had four pages which described a positive sentiment of a +3 weight, and two pages which described a negative sentiment of -1. Since a date was specified for this search, there was one article that was discarded because the crawler could not find a sensible date to use.

* crawler.log – A verbose log of every action the crawler takes. It records the following data:
  + The list of keywords, URLs, positive words, and negative words that the program is using for its crawl.
  + Every URL it visits during the course of the crawl, relevant or not.
  + Any words from the positive or negative word list that is found within a page.
  + Any instance where an article is deemed too old, too new, or that no date is found.

This file is useful to verify that the program is processing all the input files correctly, that it is following links correctly, and to understand the word and date analysis that is being performed.

**Options**

The following options are supported for the program.

* Help mode (-h, --help) – prints a concise help message that explains these options and the format of the input files, and then exits.
* Debug mode (-d, --debug) – everything that is printed to the log file (see above) will also be printed to the screen. Note that that this creates a lot of output, most of which will not be relevant to the crawl itself.
* Quiet mode (-q, --quiet) – suppresses all output except the ending summary. Useful for long crawls where lots of output is undesirable.
* Ignore mode (-i, --ignore) – ignores missing dates from relevant articles. Assumes that articles with missing dates meet the date criteria, instead of discarding articles for which the crawler cannot find a date for. Useful for websites with unorthodox methods of storing publication dates.

For example, to use ignore mode, the following command is used:

perl crawler.pl <input> <output> -i

Options can be combined, but quiet mode takes precedence over debug mode (if both options are used). To use ignore mode and quiet mode:

perl crawler.pl <input> <output> -i –q

Long-form versions of the modes can be used for clarity. The following command is equivalent to the above command:

perl crawler.pl <input> <output> --ignore --quiet