1. I had to change how politeness was implemented. I was running into some issues with accessing the pages when recrawling. Previously I just had the program wait 0.2 seconds between each request. Now if it fails it waits 3 seconds then tries again.

The library I was using to strip the webpages put the title of the page with the rest of the information in the documents when I called its “got\_text()” function. I refactered to remove the title from the html code before writing the text to file.

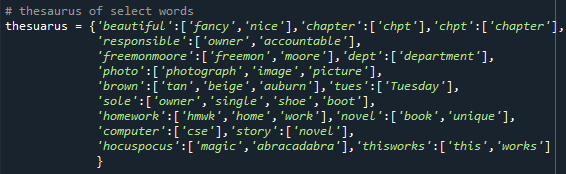
I changed my index from being just the name of the file to a tuple containing the URL, the name of the file, and the title of the file.

I also changed my interpretation of a word. Previously I had decided that hyphenated words were words, I changed to splitting hyphenated words into their constituent parts.

I had issues with handling none ASCII characters, which I have resolved.

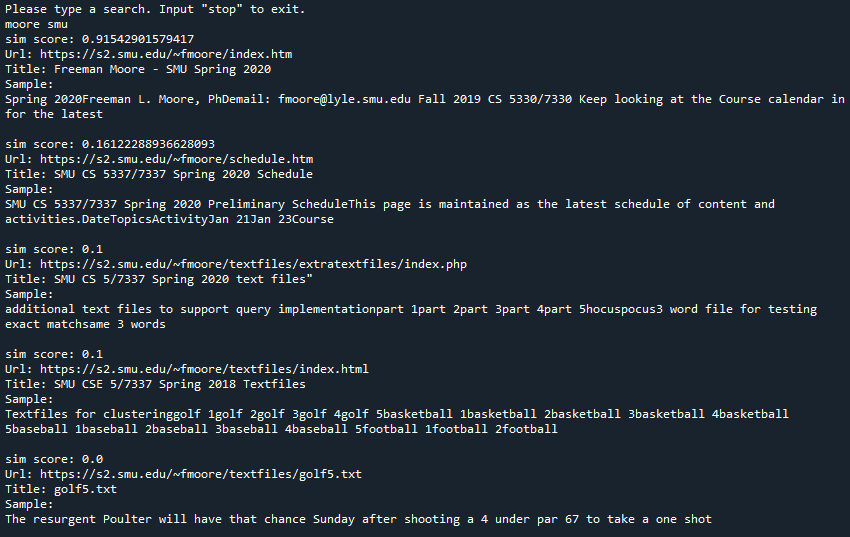
1. If a user enters a word which does not appear in the dictionary my program will return a similarity score of 0 for all the documents in the collection

I implemented the thesaurus using a dictionary as seen below:



If a word has a synonym, its synonyms are appended to the query

1. I used the suggested lnc.ltc
3. moore smu:



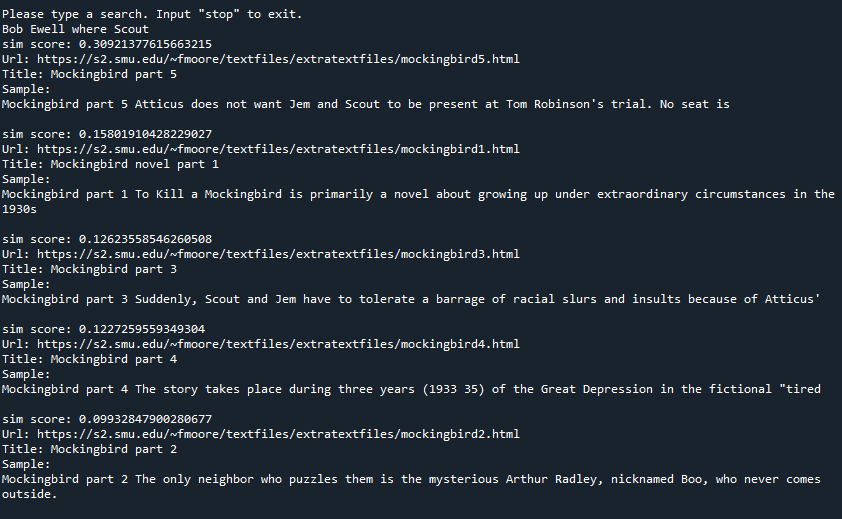
The first page index.htm has lots of Moore and smu things to make hits on.

The second, schedule.htm, contains a couple SMU’s and has SMU in the title.

The third and fourth only have SMU in the titles.

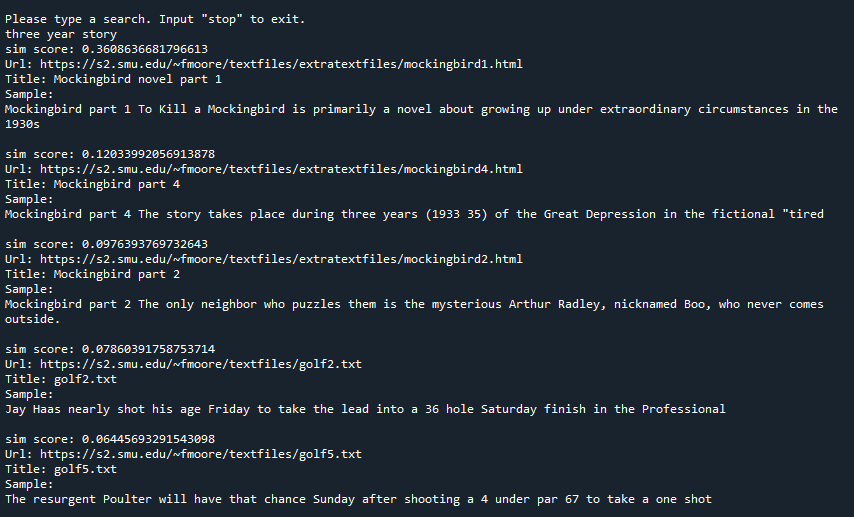
There is not a fifth relevant document so a random one is chosen.

1. Bob Ewell where Scout



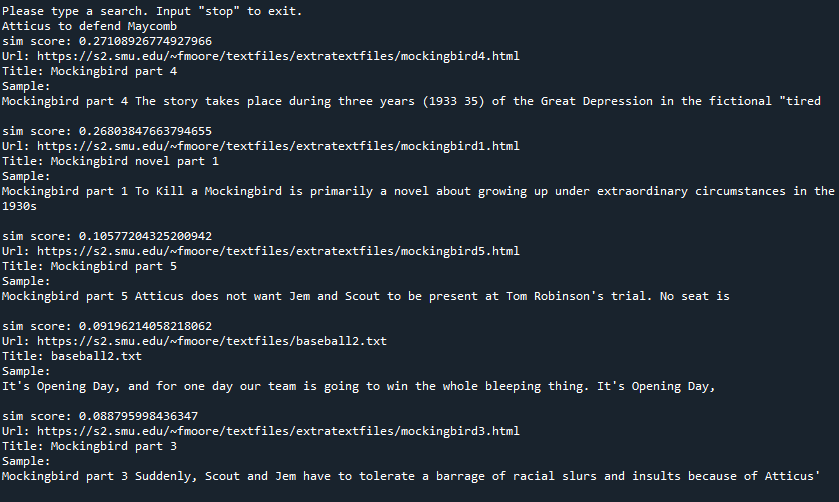
This returns the 5 parts of mockingbird. Part 5 is the highest as it mentions Bob Ewell and Scout many times. Where as the other only mention Scout in decreasing order.

1. three year story



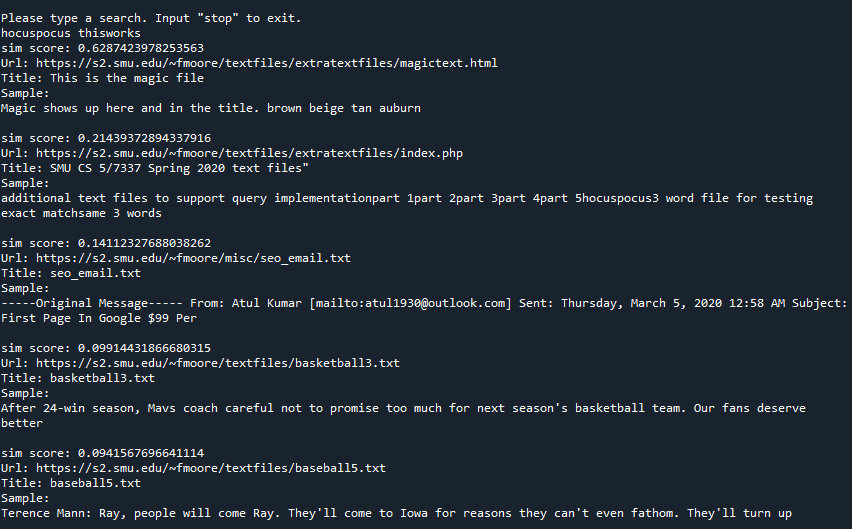
It returns Mockingbird parts 1, 4 and 2 as well as golf2 and golf5. Looking at these pieces it makes since. Mockingbird 1 has instances of each of the query words. And the rest have a handful of instances of one or more of the query words.

1. Atticus to defend Maycomb



This returns Parts 4,1,5, and 3 of Mockingbird and baseball2. Mockingbird ones makes since as Maycomb and Atticus are characters in Mockingbird. Baseball2 contains several “to” and defending, which likely gets stemmed to defend by the stemmer.

1. hocuspocus thisworks



Returns magictext.html, index.php, seo\_email.txt, basketball3.txt, and basketball5.txt. the magic text file makes since as “hocus-pocus” has a thesaurus entry with “magic”. Index.php contains the phase “hocuspocus”. Seo\_email.txt has many instances of “this” and “work” which are thesaurus entries for “thisworks”. Basketball3 and baseball5 just contain “this” a few times.