Assignment 3

SJSU Students

Exercise Part

4.1) In Section 4.3 we introduced the concept of the per-term index as a means to improve the index's random access performance. Suppose the posting lists for some term consists of 96 million postings each of which is 2 bytes long. In order to carry out single random access into the term's postings list, the search engine needs to perform two disk read operations:

1. Loading the per-term index into RAM
2. Loading a block B of postings into RAM, where B is identified by means of binary search on the list of synchronization points.

Let us call the number of postings per synchronization point the granularity of the per-term index. For the above access pattern, what is the optimal granularity (i.e., the one that minimizes the disk I/O)? What is the total number of bytes read from disk?

Cost function of disk I/O = Per-Term index length + block length
and Per-Term index length = posting list length/(block length)
Cost function of disk I/O = posting list length/(block length) + block length
x = block length
f(x) = (96 * 10^6 * 2) / x + x
df/dx = 1 - 192*10^6/x^2
In order to minimize df/dx, set it equal to 0 and solve for x.
0 = 1 - 192*10^6/x^2
x^2 = 192*10^6
x = 13856.4064606 bytes

The optimum length of the block should be 13856 bytes since we have to round to the nearest 2 byte increment and f(13856) is smaller than f(13858). Therefore each posting list will contain 6928 postings and each block should contain 6928 postings. The last block will have a little less at 5632 postings. The optimal granularity of the problem having a per-term index length of 13856 bytes and a block length of 13856 bytes. The total number of bytes read from disk is 13856 * 2 = 27712 bytes.

4.3) Building an inverted index is essentially a sorting process. The lower bound for every general-purpose sorting algorithm is Ω(n log(n)). However, the merge-based index construction method from Section 4.5.3 has a running time that is linear in the size of the collection (see Table 4.7, page 130). Find at least two places where there is a hidden logarithmic factor.

(a) From the while loop at line 5 in figure 4.13, the algorithm checks for the smallest value (in alphabetical order) among all the partitions. It starts from the first term and parses through the entire set of each partition for the least term and adds the least term to the final index.

(b) There should be an optimum amount of memory space allocated to the indexing process. The performance of the algorithm suffers severely if there is less disk space allocated to the indexing process.

(c) In line 12 of figure 4.12, there is a call to sort an in-memory dictionary. The time complexity of sorting dictionary entries in lexicographical order is n*log(n)

4.4) In the algorithm shown in Figure 4.12, the memory limit is expressed as the number of postings that can be stored in RAM. What is the assumption that justifies this definition of the memory limit? Give an example of a text collection or an application in which the assumption does not hold.

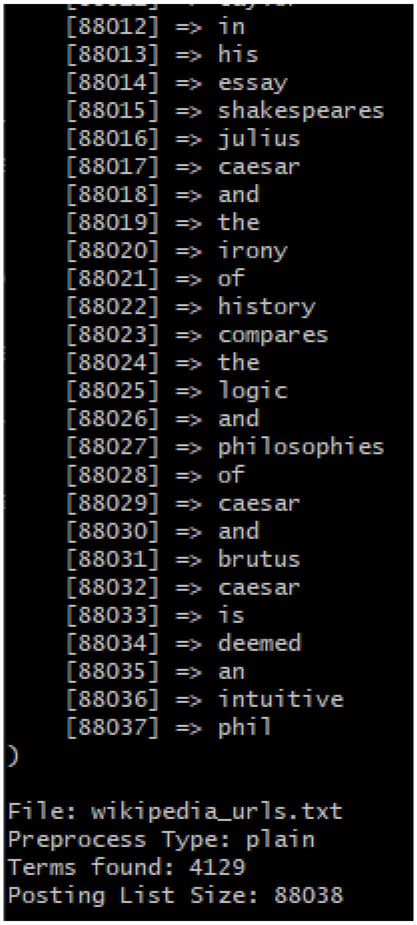
While choosing a memory limit for the number of postings, we need to consider providing the optimum amount of space for the read-ahead buffer, at least a few hundred kilobytes in each partition. It helps to keep the number of disk seeks small.

The algorithm increments the memoryConsumption variable by 1 for each term encountered until it reaches memoryLimit in figure 4.12 in the textbook. This step in the algorithm highlights the assumption: the amount of memory that each posting takes is an equal amount of memory.

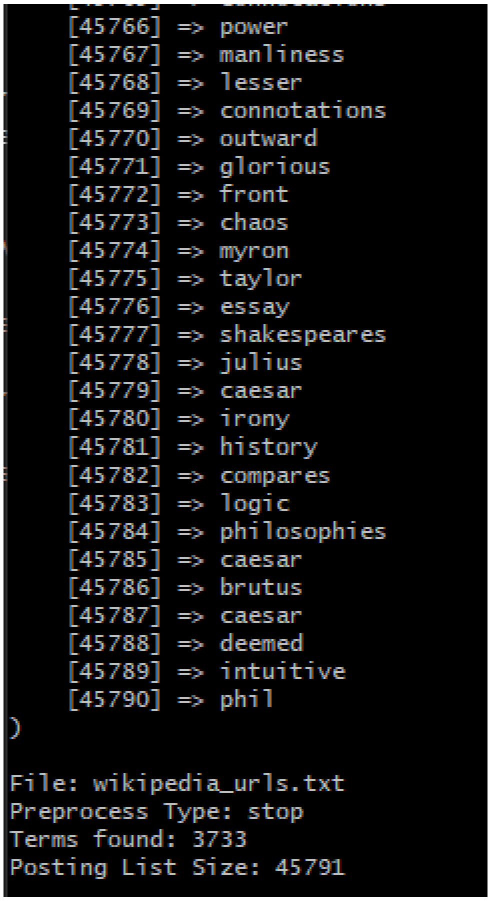
An example where this assumption does not hold is where the postings have the form (docid, tf, <positions>). In this case, each term is not stored in a fixed size. If each term’s docid and tf take 1 byte and each position takes 1 byte then duplicate words in the same document will have different memory allocations. For instance, the document containing “dog cat dog” would not take 3*1 bytes. It would take 3 (new tuple of dog) + 3 (new tuple of cat) + 1 (add one more to the existing posting list of dog) bytes to store in a posting list.

Programming Part Experiments:

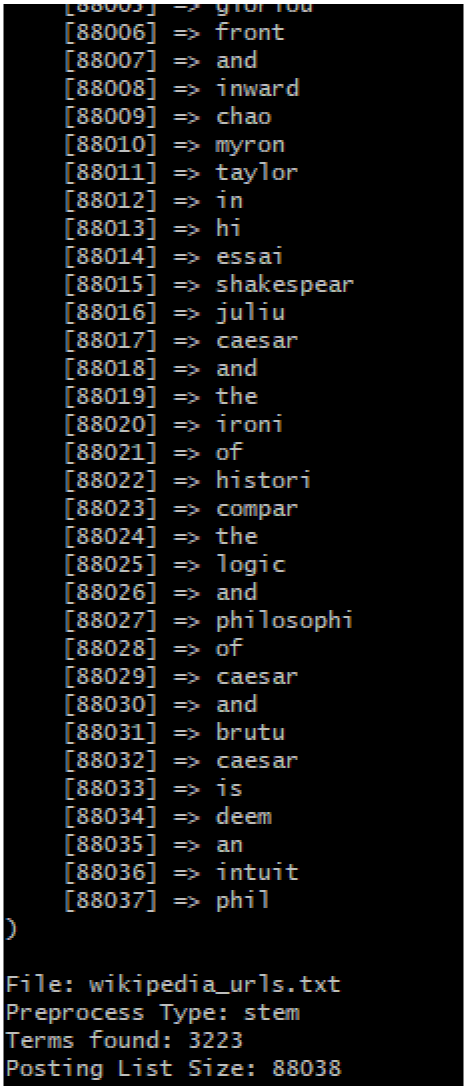
Experiment 0: Wiki Articles based on Various Shakespeare Plays



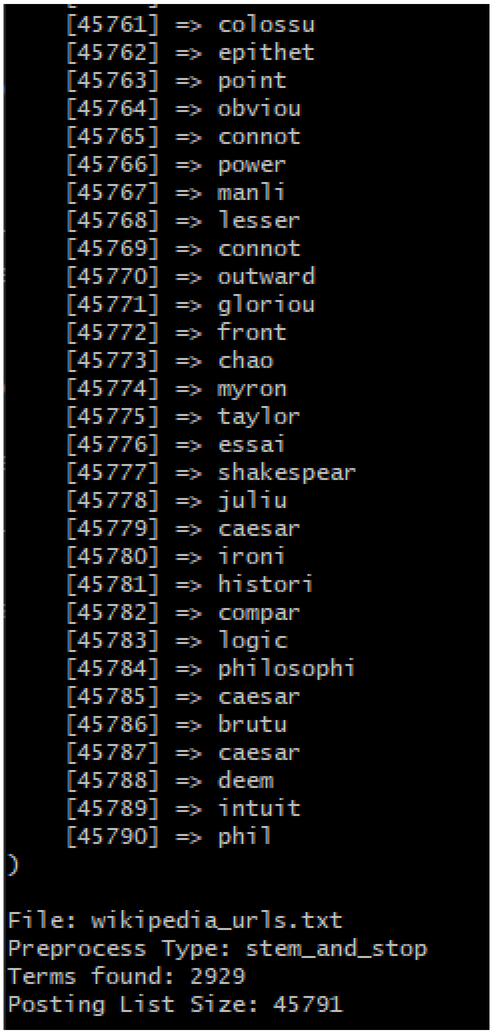
```
    [88012] => in
    [88013] => his
    [88014] => essay
    [88015] => shakespeares
    [88016] => julius
    [88017] => caesar
    [88018] => and
    [88019] => the
    [88020] => irony
    [88021] => of
    [88022] => history
    [88023] => compares
    [88024] => the
    [88025] => logic
    [88026] => and
    [88027] => philosophies
    [88028] => of
    [88029] => caesar
    [88030] => and
    [88031] => brutus
    [88032] => caesar
    [88033] => is
    [88034] => deemed
    [88035] => an
    [88036] => intuitive
    [88037] => phil
)

File: wikipedia_urls.txt
Preprocess Type: plain
Terms found: 4129
Posting List Size: 88038
```



```
    [45766] => power
    [45767] => manliness
    [45768] => lesser
    [45769] => connotations
    [45770] => outward
    [45771] => glorious
    [45772] => front
    [45773] => chaos
    [45774] => myron
    [45775] => taylor
    [45776] => essay
    [45777] => shakespeares
    [45778] => julius
    [45779] => caesar
    [45780] => irony
    [45781] => history
    [45782] => compares
    [45783] => logic
    [45784] => philosophies
    [45785] => caesar
    [45786] => brutus
    [45787] => caesar
    [45788] => deemed
    [45789] => intuitive
    [45790] => phil
)

File: wikipedia_urls.txt
Preprocess Type: stop
Terms found: 3733
Posting List Size: 45791
```

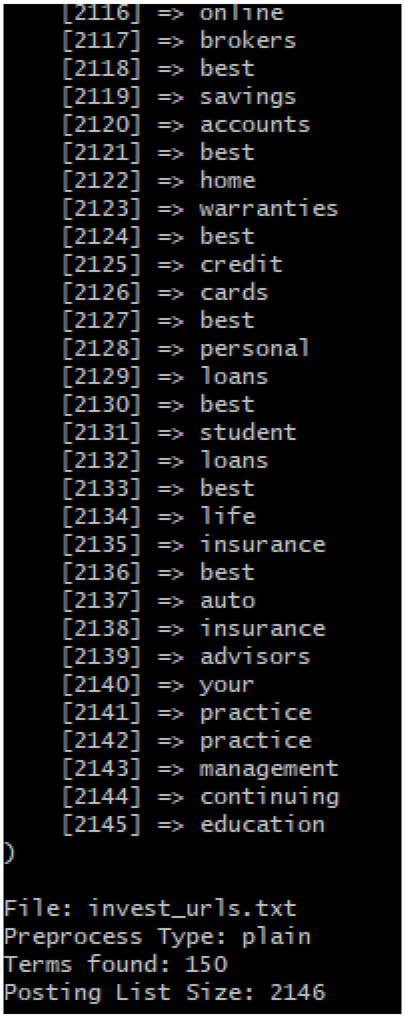


```
    [88006] => front
    [88007] => and
    [88008] => inward
    [88009] => chao
    [88010] => myron
    [88011] => taylor
    [88012] => in
    [88013] => hi
    [88014] => essai
    [88015] => shakespear
    [88016] => juliu
    [88017] => caesar
    [88018] => and
    [88019] => the
    [88020] => ironi
    [88021] => of
    [88022] => histori
    [88023] => compar
    [88024] => the
    [88025] => logic
    [88026] => and
    [88027] => philosophi
    [88028] => of
    [88029] => caesar
    [88030] => and
    [88031] => brutu
    [88032] => caesar
    [88033] => is
    [88034] => deem
    [88035] => an
    [88036] => intuit
    [88037] => phil
)

File: wikipedia_urls.txt
Preprocess Type: stem
Terms found: 3223
Posting List Size: 88038
```

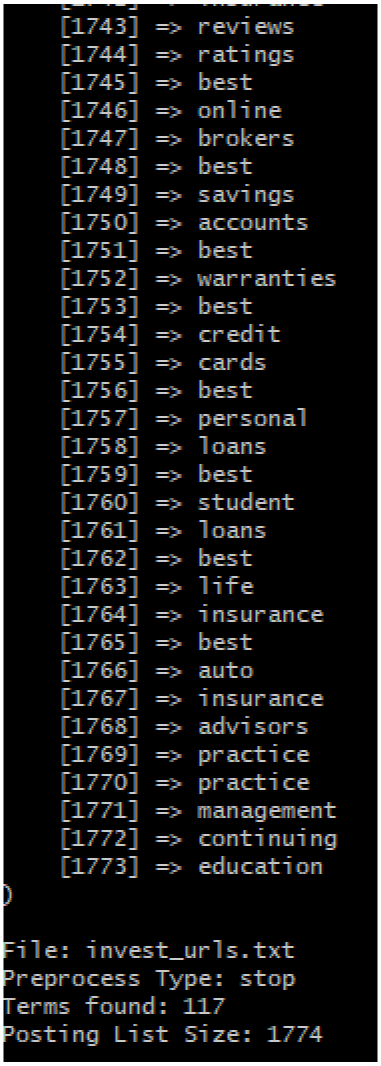


```
    [45761] => colossu
    [45762] => epithet
    [45763] => point
    [45764] => obviou
    [45765] => connot
    [45766] => power
    [45767] => manli
    [45768] => lesser
    [45769] => connot
    [45770] => outward
    [45771] => gloriou
    [45772] => front
    [45773] => chao
    [45774] => myron
    [45775] => taylor
    [45776] => essai
    [45777] => shakespear
    [45778] => juliu
    [45779] => caesar
    [45780] => ironi
    [45781] => histori
    [45782] => compar
    [45783] => logic
    [45784] => philosophi
    [45785] => caesar
    [45786] => brutu
    [45787] => caesar
    [45788] => deem
    [45789] => intuit
    [45790] => phil
)

File: wikipedia_urls.txt
Preprocess Type: stem_and_stop
Terms found: 2929
Posting List Size: 45791
```

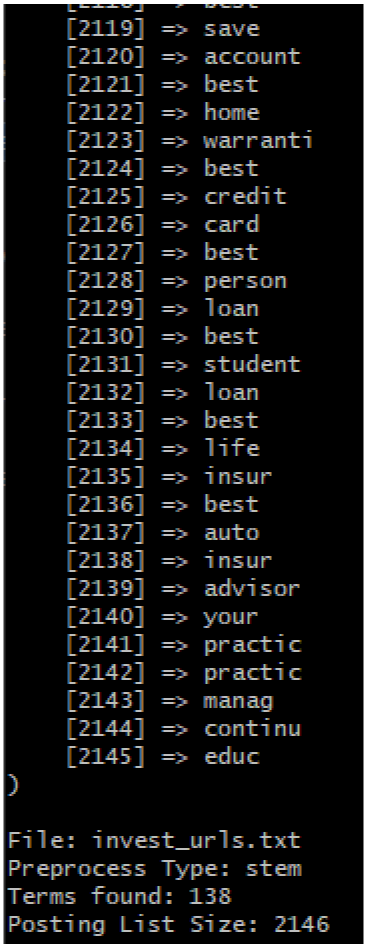
Experiment 1: investopedia Articles



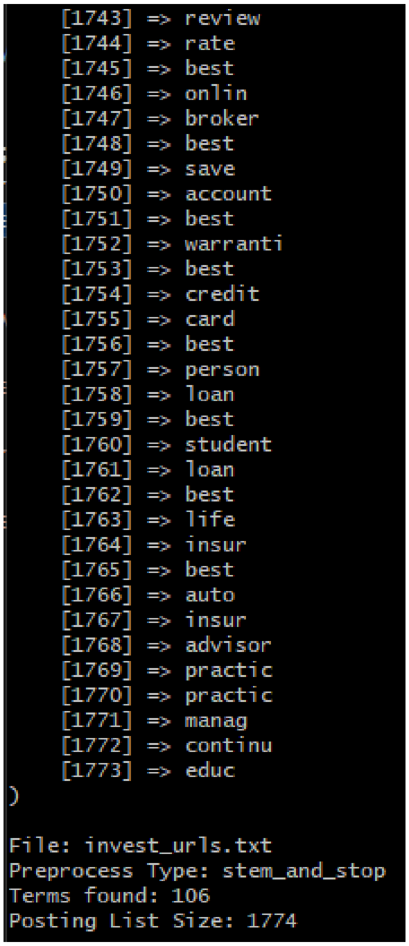
```
    [2116] => online
    [2117] => brokers
    [2118] => best
    [2119] => savings
    [2120] => accounts
    [2121] => best
    [2122] => home
    [2123] => warranties
    [2124] => best
    [2125] => credit
    [2126] => cards
    [2127] => best
    [2128] => personal
    [2129] => loans
    [2130] => best
    [2131] => student
    [2132] => loans
    [2133] => best
    [2134] => life
    [2135] => insurance
    [2136] => best
    [2137] => auto
    [2138] => insurance
    [2139] => advisors
    [2140] => your
    [2141] => practice
    [2142] => practice
    [2143] => management
    [2144] => continuing
    [2145] => education
)

File: invest_urls.txt
Preprocess Type: plain
Terms found: 150
Posting List Size: 2146
```



```
    [1743] => reviews
    [1744] => ratings
    [1745] => best
    [1746] => online
    [1747] => brokers
    [1748] => best
    [1749] => savings
    [1750] => accounts
    [1751] => best
    [1752] => warranties
    [1753] => best
    [1754] => credit
    [1755] => cards
    [1756] => best
    [1757] => personal
    [1758] => loans
    [1759] => best
    [1760] => student
    [1761] => loans
    [1762] => best
    [1763] => life
    [1764] => insurance
    [1765] => best
    [1766] => auto
    [1767] => insurance
    [1768] => advisors
    [1769] => practice
    [1770] => practice
    [1771] => management
    [1772] => continuing
    [1773] => education
)

File: invest_urls.txt
Preprocess Type: stop
Terms found: 117
Posting List Size: 1774
```

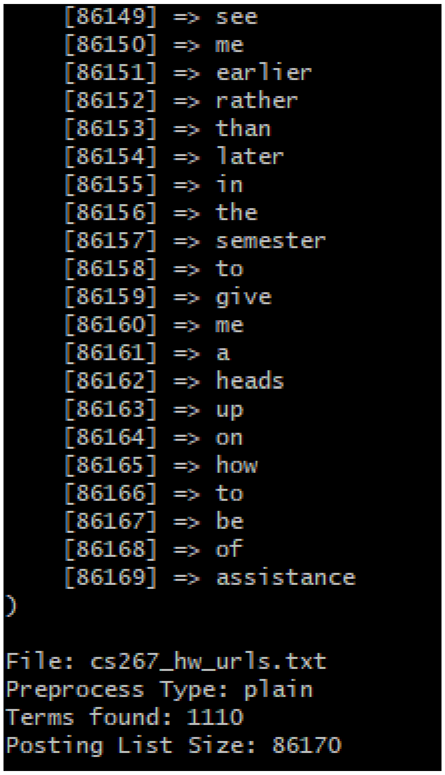


```
    [2119] => save
    [2120] => account
    [2121] => best
    [2122] => home
    [2123] => warranti
    [2124] => best
    [2125] => credit
    [2126] => card
    [2127] => best
    [2128] => person
    [2129] => loan
    [2130] => best
    [2131] => student
    [2132] => loan
    [2133] => best
    [2134] => life
    [2135] => insur
    [2136] => best
    [2137] => auto
    [2138] => insur
    [2139] => advisor
    [2140] => your
    [2141] => practic
    [2142] => practic
    [2143] => manag
    [2144] => continu
    [2145] => educ
)

File: invest_urls.txt
Preprocess Type: stem
Terms found: 138
Posting List Size: 2146
```

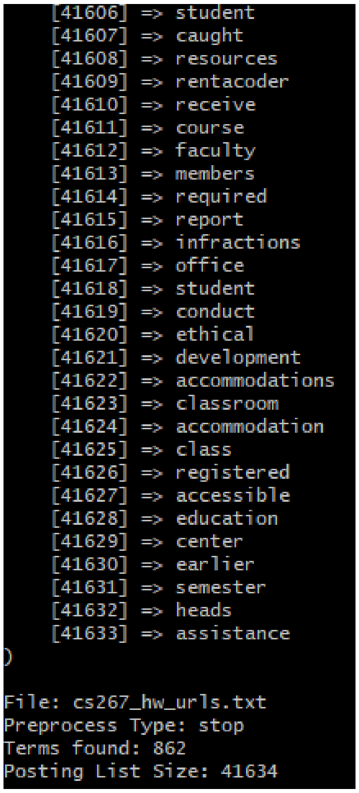


```
    [1743] => review
    [1744] => rate
    [1745] => best
    [1746] => onlin
    [1747] => broker
    [1748] => best
    [1749] => save
    [1750] => account
    [1751] => best
    [1752] => warranti
    [1753] => best
    [1754] => credit
    [1755] => card
    [1756] => best
    [1757] => person
    [1758] => loan
    [1759] => best
    [1760] => student
    [1761] => loan
    [1762] => best
    [1763] => life
    [1764] => insur
    [1765] => best
    [1766] => auto
    [1767] => insur
    [1768] => advisor
    [1769] => practic
    [1770] => practic
    [1771] => manag
    [1772] => continu
    [1773] => educ
)

File: invest_urls.txt
Preprocess Type: stem_and_stop
Terms found: 106
Posting List Size: 1774
```

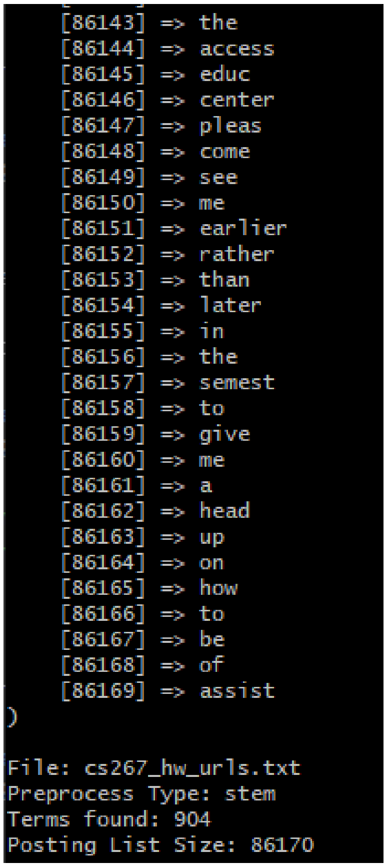
Experiment 2: CS267 Course site parsing



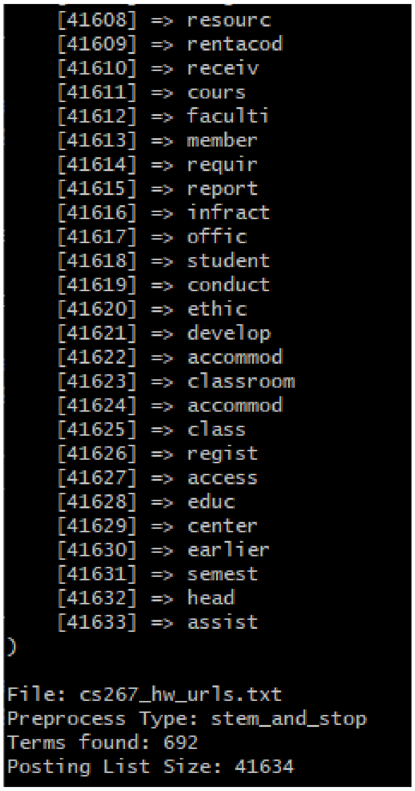
```
    [86149] => see
    [86150] => me
    [86151] => earlier
    [86152] => rather
    [86153] => than
    [86154] => later
    [86155] => in
    [86156] => the
    [86157] => semester
    [86158] => to
    [86159] => give
    [86160] => me
    [86161] => a
    [86162] => heads
    [86163] => up
    [86164] => on
    [86165] => how
    [86166] => to
    [86167] => be
    [86168] => of
    [86169] => assistance
)

File: cs267_hw_urls.txt
Preprocess Type: plain
Terms found: 1110
Posting List Size: 86170
```



```
    [41606] => student
    [41607] => caught
    [41608] => resources
    [41609] => rentacoder
    [41610] => receive
    [41611] => course
    [41612] => faculty
    [41613] => members
    [41614] => required
    [41615] => report
    [41616] => infractions
    [41617] => office
    [41618] => student
    [41619] => conduct
    [41620] => ethical
    [41621] => development
    [41622] => accommodations
    [41623] => classroom
    [41624] => accommodation
    [41625] => class
    [41626] => registered
    [41627] => accessible
    [41628] => education
    [41629] => center
    [41630] => earlier
    [41631] => semester
    [41632] => heads
    [41633] => assistance
)

File: cs267_hw_urls.txt
Preprocess Type: stop
Terms found: 862
Posting List Size: 41634
```

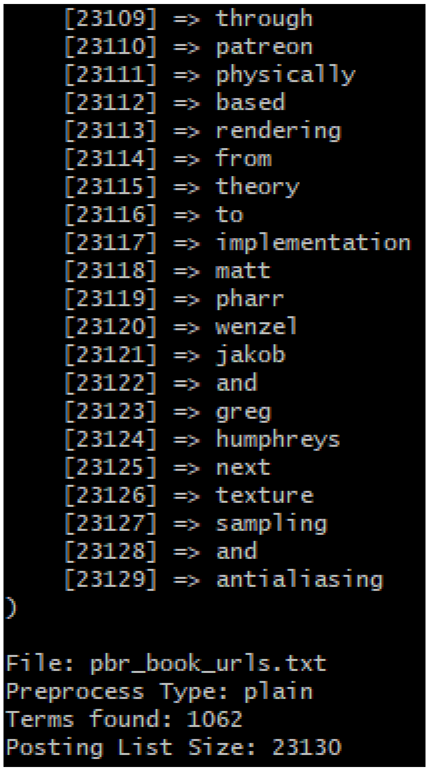


```
    [86143] => the
    [86144] => access
    [86145] => educ
    [86146] => center
    [86147] => pleas
    [86148] => come
    [86149] => see
    [86150] => me
    [86151] => earlier
    [86152] => rather
    [86153] => than
    [86154] => later
    [86155] => in
    [86156] => the
    [86157] => semest
    [86158] => to
    [86159] => give
    [86160] => me
    [86161] => a
    [86162] => head
    [86163] => up
    [86164] => on
    [86165] => how
    [86166] => to
    [86167] => be
    [86168] => of
    [86169] => assist
)

File: cs267_hw_urls.txt
Preprocess Type: stem
Terms found: 904
Posting List Size: 86170
```

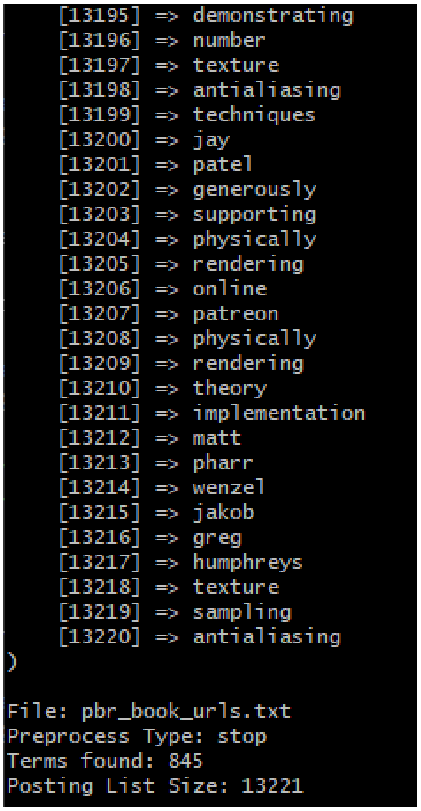


```
    [41608] => resourc
    [41609] => rentacod
    [41610] => receiv
    [41611] => cours
    [41612] => faculti
    [41613] => member
    [41614] => requir
    [41615] => report
    [41616] => infract
    [41617] => offic
    [41618] => student
    [41619] => conduct
    [41620] => ethic
    [41621] => develop
    [41622] => accommod
    [41623] => classroom
    [41624] => accommod
    [41625] => class
    [41626] => regist
    [41627] => access
    [41628] => educ
    [41629] => center
    [41630] => earlier
    [41631] => semest
    [41632] => head
    [41633] => assist
)

File: cs267_hw_urls.txt
Preprocess Type: stem_and_stop
Terms found: 692
Posting List Size: 41634
```

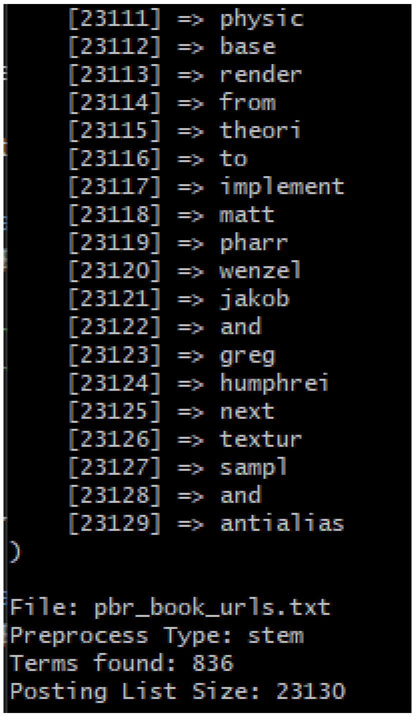
Experiment 3: Online HTML-based Textbook



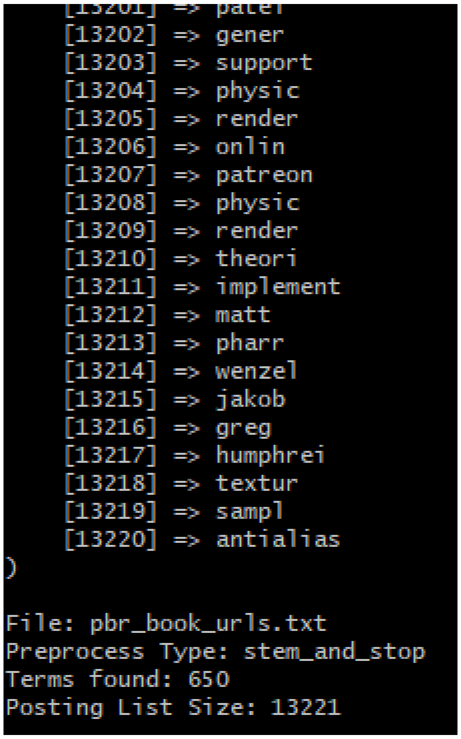
```
    [23109] => through
    [23110] => patreon
    [23111] => physically
    [23112] => based
    [23113] => rendering
    [23114] => from
    [23115] => theory
    [23116] => to
    [23117] => implementation
    [23118] => matt
    [23119] => pharr
    [23120] => wenzel
    [23121] => jakob
    [23122] => and
    [23123] => greg
    [23124] => humphreys
    [23125] => next
    [23126] => texture
    [23127] => sampling
    [23128] => and
    [23129] => antialiasing
)

File: pbr_book_urls.txt
Preprocess Type: plain
Terms found: 1062
Posting List Size: 23130
```



```
    [13195] => demonstrating
    [13196] => number
    [13197] => texture
    [13198] => antialiasing
    [13199] => techniques
    [13200] => jay
    [13201] => patel
    [13202] => generously
    [13203] => supporting
    [13204] => physically
    [13205] => rendering
    [13206] => online
    [13207] => patreon
    [13208] => physically
    [13209] => rendering
    [13210] => theory
    [13211] => implementation
    [13212] => matt
    [13213] => pharr
    [13214] => wenzel
    [13215] => jakob
    [13216] => greg
    [13217] => humphreys
    [13218] => texture
    [13219] => sampling
    [13220] => antialiasing
)

File: pbr_book_urls.txt
Preprocess Type: stop
Terms found: 845
Posting List Size: 13221
```



```
    [23111] => physic
    [23112] => base
    [23113] => render
    [23114] => from
    [23115] => theori
    [23116] => to
    [23117] => implement
    [23118] => matt
    [23119] => pharr
    [23120] => wenzel
    [23121] => jakob
    [23122] => and
    [23123] => greg
    [23124] => humphrei
    [23125] => next
    [23126] => textur
    [23127] => sampl
    [23128] => and
    [23129] => antialias
)

File: pbr_book_urls.txt
Preprocess Type: stem
Terms found: 836
Posting List Size: 23130
```



```
    [13202] => gener
    [13203] => support
    [13204] => physic
    [13205] => render
    [13206] => onlin
    [13207] => patreon
    [13208] => physic
    [13209] => render
    [13210] => theori
    [13211] => implement
    [13212] => matt
    [13213] => pharr
    [13214] => wenzel
    [13215] => jakob
    [13216] => greg
    [13217] => humphrei
    [13218] => textur
    [13219] => sampl
    [13220] => antialias
)

File: pbr_book_urls.txt
Preprocess Type: stem_and_stop
Terms found: 650
Posting List Size: 13221
```

The plain argument is our base and we will compare all changes to that.

Preprocess Type: plain
Terms found: 4127
Posting List Size: 88017

Stemming seems to have the effect on reducing dictionary size. Since there are many variations of english words that share the same root word, the amount of keys gets reduced. However, stemming has no effect on the size of the overall index because it does not remove any postings, instead they just get combined into one term entry.

Preprocess Type: stem
Terms found: 3221
Posting List Size: 88017

Stopping has the effect of reducing the dictionary size as well. Since there are many stop words contained in english paragraphs, large posting lists get removed. Therefore, there is a large effect on the reduction of the posting list size.

Preprocess Type: stop
Terms found: 3731
Posting List Size: 45780

Stemming and stopping seems to have the most effect on reducing the dictionary size and posting list size since both techniques are applied and both methods of reduction are also applied.

Preprocess Type: stem_and_stop
Terms found: 2927
Posting List Size: 45780

Calling CrawlConstants::DESCRIPTION had different output sizes depending on the website that the experiment's urls were based on. Simple-format HTML pages like wiki and Experiment 2 & 3 had the posting list Size increased greatly compared to the likes of Experiment 1 & other Wiki-like websites (Encyclopedia,info please, etc.). Experiment 1 had the article name and categorical elements in the Description's diction but did not dwell in the subject matter of topic statements like Experiment 3 with its textbook format. This resulted in the posting list size to be smaller than in other experiments but had a more relevant dictionary size.

Something to keep in mind is that Experiment 1 had urls of varying topics likely to proportionally have more distinct terms to be found instead of Experiment 0's general topic of Shakespere. However, given the big quantity of words picked up by the Summarizer in Experiment 0, the proportional difference isn't shown in this instance and would be relevant with more experiments.

hike stems to hik
hikes and hiked also stem to hik

query = "hike"
doc1 = "I love to hike"
doc2 = "Taking hikes is a great way to relax"
doc3 = "The boy hiked up his sweatpants"

doc1 and doc2 are relevant to the query and doc3 is irrelevant.

Stemming increases recall because without stemming you would get a recall of 1/2. With stemming, we match all the docs, giving us a recall of 2/2.

In this same example precision is also decreased. Since without stemming you get a precision of 1/1. With stemming, precision is decreased since all of the documents are returned and precision drops to 2/3.