COEN 272 – Project 1

- ✓ **Web Crawling**
- ✓ **Content Processing**
- ✓ **Zipf's Law**

By

PRIYANKA RAO AND KAVYA GAUTAM

1. INTRODUCTION 3

1.1 ABOUT THE PROJECT 3

MAIN COMPONENTS: 3

2. IMPLEMENTATION DETAILS 3

2.1 WEB CRAWLER 3

FLOW OF THE IMPORTANT FUNCTIONS: 3

2.2 CONTENT PROCESSOR 3

FLOW OF THE IMPORTANT FUNCTIONS: 4

CHOICE OF NOISE REDUCTION TECHNIQUE 4

CHOICE OF TECHNIQUE FOR EVALUATING THE NOISE REDUCTION PERFORMED 5

ANALYSIS OF THE NOISE REDUCTION TECHNIQUE BASED ON THE CHOSEN EVALUATION METRICS 5

CHALLENGES FACED 5

2.3 ZIPF'S LAW – WORD FREQUENCY COUNT 1

APPENDIX 1

SUBMISSION DETAILS: 1

OUTPUTS, SCREENSHOTS AND PLOTS 2

REFERENCES 1

1. INTRODUCTION

1.1 ABOUT THE PROJECT

MAIN COMPONENTS:

The project is divided into three main components:

- Web Crawler
- Content processor and
- Word frequencies – Zipf's law

The web crawler and content processor sections are built in Java and Zipf's distribution is constructed using Excel

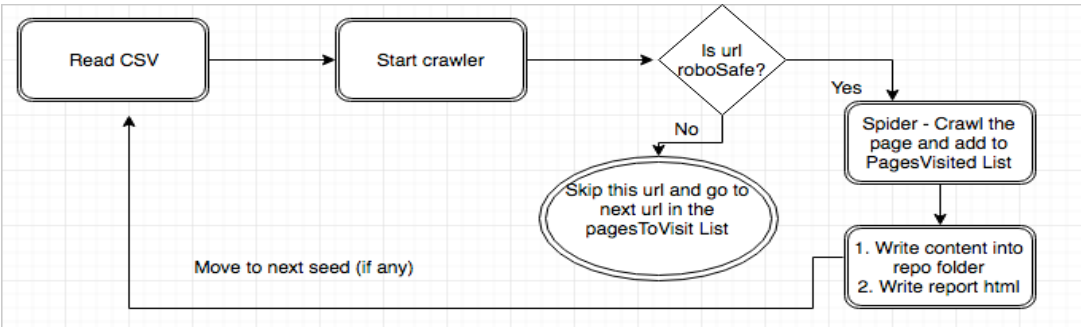
2. IMPLEMENTATION DETAILS

2.1 WEB CRAWLER

The Web crawler obeys the politeness policies of the host url's robots.txt, before visiting any page

FLOW OF THE IMPORTANT FUNCTIONS:

1. readcsv() - Reads the specification file one line at a time and starts the crawler for the seed if the seed is a valid and allowed page
2. robotSafe(url) – checks for each link, if the link is disallowed in robots.txt file of the main website and returns true if it does obey.
3. startCrawler() – sets the crawler on, based on some safety flags, domain restriction flags etc
4. Spider(url, domainword) – Connects to URL, fetches details like status code, number of links and images found etc., and sets of to download the content – Uses Jsoup
5. writeContent(document) – Creates Repository folder and other files and writes the downloaded content to these files
6. writeReportHtml(url, path) – Makes an entry into report.html, every time a page is successfully visited



2.2 CONTENT PROCESSOR

The methods related to Contents processing are also part of Webcrawler.java file.

FLOW OF THE IMPORTANT FUNCTIONS:

1. contentProcessor() - Reads in all the files downloaded from the repository folder, removes the basic noise tags using jsoup selectors, then applies Link-Text analysis to remove further noise. After this step, the content processor also utilizes BlockSizeRatio techniques through using NoiseRemoval class to extract the main and longest content of the web page. It then writes the processed content into an output "crawlercontent.html" file as text
   Credits for most parts of NoiseRemoval methods: https://github.com/vikasing/NiceText
2. writeTermFrequency() – Computes the word frequency count from all the words generated in the above content file (making sure the whitespaces and special characters are removed) and generates the output text file with the words and their counts. This also utilizes some other helper functions that count the word frequency and arrange them in descending order of frequency

CHOICE OF NOISE REDUCTION TECHNIQUE

1.*Pre-Processing:*

Step 1: As a preprocessing step we removed visually noticeable noise like navigation, header, footer, advertisements etc., because they are unrelated to the main content of the page

Step 2: Removed script code, images and video files if any, form elements, buttons, search box, etc. Also removed empty 'div' blocks that mainly usually contain Ad related content. These elements will be of minimal or no use when it comes to our eventual goal of information retrieval

2.Text-Link Analysis:

Step 3: The first technique is to analyze the links (hrefs) in each element of the DOM structure, and if the length of the text of the link is <=4, it can be safely ignored

Example: In the following example, the links highlighted in red are of no importance to the main content and hence, their anchor text can be safely discarded.

Encyclopedia Britannica's.[14] Criticisms of Wikipedia include claims that it exhibits systemic bias, presents a mixture of "truths, half truths, and some falsehoods",[15] and that in controversial topics it is subject to manipulation and spin.[16]

Step 4: The second technique is to traverse through the DOM structure using Jsoup, if a link(a tag) is encountered, step back and look at its immediate parent. Now find the child size of the parent. If the size happens to be 1, it means that the child href is a solidary link which is part of listed links in the non-content related blocks. So this link can be removed. Rest of the links will be retained for their anchor text.

Example: The following shows a sample of such solidary link that can be discarded

Step 5: Check for each a-href found on the doc, if the anchor text of the link contains the title of the page. If it does, surpass the previous conditions on links and retain the anchor text for this

Example: Here, the wiki page for fish, has some links within named fish, so they will be retained even though the length of the anchor text is <= 4

A **fish** is any member of a paraphyletic group of organisms tha
craniate animals that lack limbs with digits. Included in this de
and cartilaginous and bony fish, as well as various extinct rela

3.Additional Processing:

Step 6: The additional processing step uses the getText() method of NoiseRemoval class. It utilizes helper methods to traverse through the DOM tree and gauze the size of the nodes and their parents and consider them only if the size of the word present >= 5. Then it uses another method to compute the block-size-ratio by normalizing each block size by the average block sizes of all the element. and if this ratio is less than a preset ratio of 0.1, the corresponding elements are ignored. This technique works well for web pages that have large main contents.

CHOICE OF TECHNIQUE FOR EVALUATING THE NOISE REDUCTION PERFORMED

- Our primary choice of technique for evaluating the noise reduction performed is Cosine similarity. We manually gathered visually relevant content from each downloaded file to one big Gold Standard Html and computed the word frequency count on it. We also generate WFC on the processed content Html document and then compute cosine similarity between these two text files of words-frequencies.
    - Cosine similarity between the two WFC docs is computed by first computing the intersection words between d1 and d2 and then the dot product and finally the product of sqrt(sum of freqs) for both documents
- We also performed LCS on the two (gold and processed) documents to derive other metrics like Precision, Recall and F1 score:
    - precision = LCS (gold, extraction)/extraction-length; Recall = LCS (gold, extraction)/gold-length; F1 = LCS(g, e) / (extraction-length + gold-length – LCS)

ANALYSIS OF THE NOISE REDUCTION TECHNIQUE BASED ON THE CHOSEN EVALUATION METRICS

We analyzed the LCS technique for 25 web-pages of processed content against the Gold standard content for a wiki article on squids (Seed) and our algorithm produced the following results (without the Additional Processing step). More than 25 pages causes Java Heap error

```
Cosine Similarity: 0.9361785178380686     Recall :0.8578494623655914
Precision :0.7508966695132366             F1 :0.7543604028283693
```

We analyzed the results of Cosine Similarity after each successful step of Noise removal as follows:

1. After the Text-Link analysis step, the cosine similarity between processed and gold contents was: `0.9361785178380686`.
2. Finally, after adding an additional processing step that takes into account the block size ratio and only retains main longest text block contents, we could bring the cosine similarity up to 0.98. While this might look like a very good value, it only works well when the seed and the crawled pages are mainly articles and blogs. For generic and miscellaneous sites, this may bring down the cosine similarity(CS) value. For eg: The CS for yahoo.com seed (20 pages) came to .67 with the additional step whereas without this step, it can be 0.73

So, in conclusion, we decided to comment out the Additional processing step in order to make the algorithm a generic one. For news articles and blogs the Additional processing step will add an advantage.

CHALLENGES FACED

1. In the Noise removal stage, arriving at a generic algorithm to cancel out the noise and also perform well on wide range of websites proved harder than we expected.

2. We went though enormous number of papers and tried implementing several other techniques based on Tag-text ratio and K means clustering of the resulting ratios. But these techniques proved futile when there isn't a huge corpus of data available to the algorithm
3. Most challenging part was to traverse the DOM structure down to element and node and decide on what to retain and what to remove.

2.3 ZIPF'S LAW – WORD FREQUENCY COUNT

- We created word-frequency counts for three different crawls, each running to max page limit of 100 pages.
- Then, we generated the Probability Vs Rank graphs in MS Excel. The graphs as expected followed the Zipf's distribution.
- The frequency of any word is inversely proportional to its rank since the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc.,
- And hence the rank-frequency relation always follows the zipf's distribution no matter the corpus of words. *The output word frequencies and the graphs are shown in the appendix*

APPENDIX

SUBMISSION DETAILS:

The Zip folder contains:

1. Project folder (bin, jars, src - .java files etc)
    a. Webcrawler.java (Main) – Includes both Web crawler and Content processor and generates all output files necessary **Run this File**
    b. RobotRule.java – Helper class to Webcrawler.java
    c. EvaluatrePart2.java – Helper class to CosineSimilarity.java
    d. CosineSimilarity.java – A standalone class, when run separately with a built GoldStandard html and word frequency text file will give the evaluation metrics
    e. NoiseRemoval.java – Additional content processing step
    f. specification.txt
2. Sample Output folder - This folder has sample of already generated output
    a. Cosine similarity evaluation output sample
    b. Repository folder's downloaded html files for one seed
    c. Report.html (contains the link, local path, http status code and other details)
    d. Crawler1content.html (Processed content from all the files in repository)
    e. Crawler1output.txt (Word frequency count from the crawler1content.html)
    f. Zipfslaw.xlsx that shows the words-frequencies and their Zipf's Law graphs for 3 crawls
3. Project Report (PDF)

External Dependencies:

1. Jsoup-1.8.3.jar
2. org.apache.commons.io_2.0.1.v201105210651.jar

How to Run:

1. Import the entire project folder "272Project1" into eclipse
2. Run the Webcrawler.java file

Input:

Specification.txt file

- Each line starting with one seed. Any number of seeds can be given
- Each line can have up to 3 tokens (seed, maxpage_limit, domain_restriction)

Output:

On running the Webcrawler.java file, the program generates the following output:

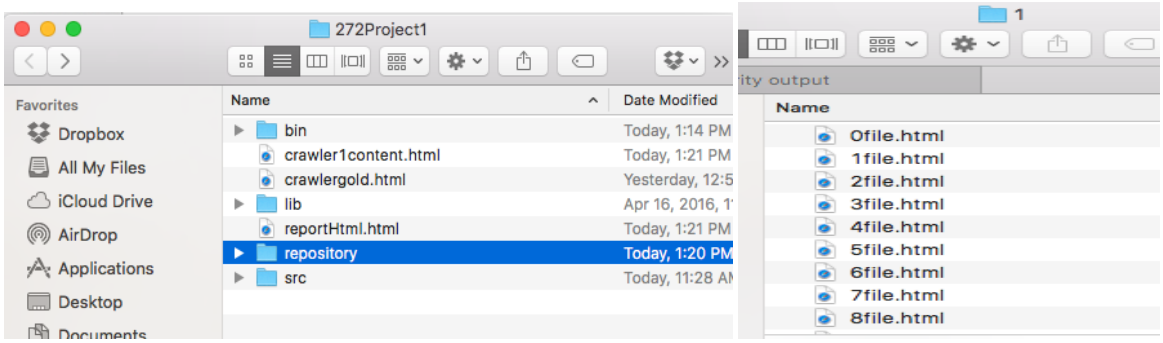
1. ONE "Repository" folder in the current project directory with numbered folders inside (numbered as per the seed). And each numbered folder will have all the links crawled and downloaded.
2. As many "crawler<seed_number>content.html" files as the number of seeds, representing the processed content of all the pages crawled from each seed
3. ONE reportHtml.html file displaying the details of crawl (with clickable links to original and local copies)
4. ONE "crawler<seed_number>output.txt" file displaying the word frequency count from its corresponding processed content file

OUTPUTS, SCREENSHOTS AND PLOTS

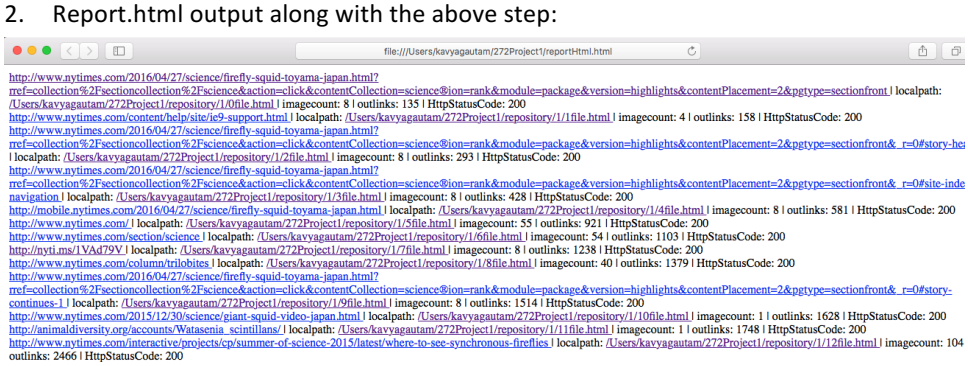
We will look at the various outputs of different parts of the project:

1. Web Crawler Repository and report html generation sample:

   Repository created in the project folder – Inside Repository is the numbered folders, one for each seed



2. Report.html output along with the above step:



3. Crawler content html after Content processing step:

file:///Users/kavyagautam/272Project1/crawler1content.html

Now Is the Time to See Squid That Glow Like Fireflies - The New York Times NYTimes.com no longer supports Internet Explorer 9 or earlier. Please upgrade your browser. LEARN MORE Â» Skip to content Skip to navigation View mobile version The New York Times ScienceÏNow Is the Time to See Squid That Glow Like Fireflies Advertisement search sponsored by http://nyti.ms/1VAd79V Advertisement Supported by Science Now Is the Time to See Squid That Glow Like Fireflies Trilobites By JOANNA KLEIN APRIL 26, 2016 Continue reading the main story Share This Page Continue reading the main story Hop on a fishing boat in Toyama Bay, Japan, in the wee hours of the morning and you may feel as if youâ€™re in a spaceship, navigating through the stars. Thatâ€™s because each year, between March and June, millions of firefly squid transform the water into a galactic landscape. Lucky for you, all you need is a reservation to come aboard, your eyes and perhaps a really good camera. The firefly squid may bring to mind a lightning bug. But the cephalopod is three inches long and flies through the sea, not the sky. And instead of a single light on its belly, it has five around each eye, three each on the tips of two of its arms and even more covering its body. The firefly squid creates light via a chemical reaction inside its body, the same way a lightning bug does. The process involves wrangling together the dynamic duo of bioluminescence, two substances called luciferin and luciferase. Luciferin sits around waiting for luciferase, an enzyme that triggers luciferin to do what it does best â€" make light. Up close, each squid looks like its own constellation of green and blue stars. Together, they form a bright blue nebula of bioluminescence, which is quite clear when the squid emerge from 1,200 feet during the spring to spawn. Firefly squid only live for a year. After making babies, they die. Why the firefly squid glows is not precisely known. Theories range from attracting mates to deterring predators, but arenâ€™t totally convincing. They also appear to lack the ability to regulate their flashes in synchronous patterns like some fireflies. Katsunori Teranishi, who studies the squidâ€™s bioluminescence at Mie University in Japan, has found that the firefly squid constantly emits a weak light with no apparent advantage. But sometimes the light shines stronger, like a flashlight, from the squidâ€™s arms. This might scare off nearby predators, but it could also attract those further away. Advertisement Continue reading the main story â€œThe light cannot actively attract a mate, communicate with a mate, and deter a predator,â€ Dr. Teranishi said. When pressed for his own hypothesis for why the squid emits light, he suggested we ask the squids themselves. The best time to observe firefly squid is in April, in Toyama City, Namerikawa and Uozu City, said Osamu Inamura, director of the Uozu Aquarium in Uozu City (where you can find firefly squid in captivity). This year tour boats depart from the Port of Namerikawa every morning at 3 a.m. from March 20 through May 8. You can catch the squid live at the only museum dedicated to them: Hotaruika Museum, also in Namerikawa. If you happen to make it to Toyama Bay this year, appreciate the fact that the firefly squid is the only squid thought to have color vision. Hereâ€™s looking at you, squid. Continue reading the main story Related Coverage Giant Squid, Elusive Creature of the Deep, Gets a Vivid Close-Up DEC. 29, 2015 Fluorescence Is Widespread in Fish, Study Finds JAN. 8, 2014 SCIENCETAKE A Sea Snail That Moves Like a Flying Insect FEB. 22, 2016 Trilobites Unearthing fascinating morsels of science. Makemake, the Moonless Dwarf Planet, Has a Moon After All APR 26 30 Years After Chernobyl Disaster, Shelter Nears Completion APR 26 Celebrate Earth Day With a 4,800-Year-Old Tree (If You Can Find It) APR 22 A Hubble Birthday Bubble in Clear View APR 21 Periods in Space Are Not That Different, Though a Bit More Complicated APR 21 Related Coverage Giant Squid, Elusive Creature of the Deep, Gets a Vivid Close-Up DEC. 29, 2015 Fluorescence Is Widespread in Fish, Study Finds JAN. 8, 2014 SCIENCETAKE A Sea Snail That Moves Like a Flying Insect FEB. 22, 2016 What's Next Loading... Go to Home Page Â» Site Index The New York Times

4. Word frequency count generated:

```
the 2885
a 1415
of 1320
to 1100
and 956
in 847
that 532
 512
is 498
this 404
on 393
for 365
by 319
it 315
at 295
from 263
with 256
as 250
are 227
but 204
they 198
you 197
its 189
an 187
s 187
page 182
not 180
new 176
share 173
he 169
```

5. Results of evaluation step – Cosine Similarity

Cosine Similarity between ./crawler1output.txt and ./crawlergoldoutput.txt is:

Webcrawler.java | specification.csv | Evaluation.java | CosineSimilarity.java | EvaluatePart2.java

Source | History

```
29
30          Set<CharSequence> intersection = getIntersection(leftVector, rightVector);
31          double dotProduct = dot(leftVector, rightVector, intersection);
32
33          double d1 = 0.0d;
            for (Integer value : leftVector.values()) {
35              d1 += Math.pow(value, 2);
36          }
37          double d2 = 0.0d;
            for (Integer value : rightVector.values()) {
39              d2 += Math.pow(value, 2);
40          }
41          double cosineSimilarity;
```

Output

Debugger Console | Webcrawler (run)

```
run:
Cosine Similarity between ./crawler1output.txt and ./crawlergoldoutput.txtis: 0.9365792915473321
BUILD SUCCESSFUL (total time: 0 seconds)
```

6. Zipf's law plot results and the top 100 words of each crawl:

A note about Domain restriction, i.e., the third token in the specification file:

We did not include domain specification in the below outputs, as sometimes, the seeds we chose will end before reaching the maximum of 100 pages, if a domain restriction is given. However, we have implemented this as both domain restriction and as a key word. If we give a specific domain as www.yahoo.com, it will only restrict the crawls to that domain. If we give any word like "trump", the crawler will effectively crawl only those links that have the word in their url path.

CRAWL 1:

Seed 1: https://www.yahoo.com/news/

Max page: 100

Domain Restriction:

Crawl only links containing the word "trump"

Top 100 words:

Terms	Frequency
the	893
to	527
a	364
in	344
of	342
and	301
	220
trump	210
for	194
that	191
on	149
this	147
yahoo	145
i	136
news	110
is	109
by	107
with	105
he	104
as	104
his	102

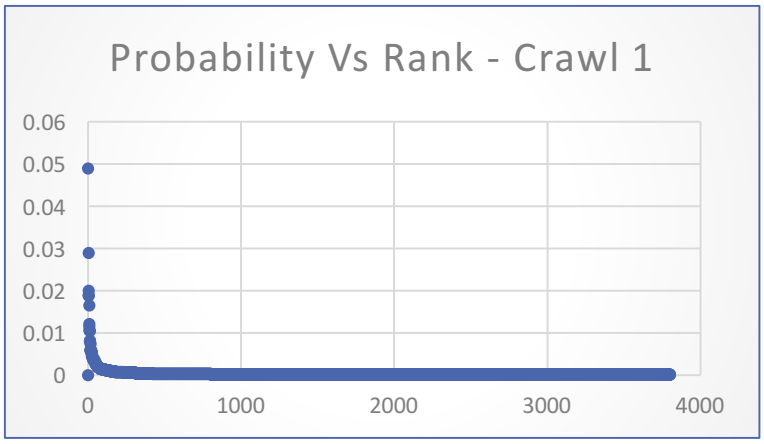
t	102
was	97
like	90
sponsored	85
have	82
don	80
it	79
at	77
from	73
ad	72
s	72
who	68
has	67
they	67
about	67
you	65
manafort	64
said	62
more	62
are	62
an	62
campaign	61

be	58
cruz	55
but	54
not	52
republican	52
donald	50
will	48
what	46
2016	45
been	45
if	45
clinton	43
press	42
new	41
were	40
all	38
or	38
one	37
presidenti	36
associated	36
sanders	36
state	35

had	34
would	34
up	33
their	31
him	31
we	31
u	31
after	30
delegates	29
years	29
than	29
now	29

april	28
free	27
gop	27
out	27
candidate	27
california	27
finance	27
trumps	26
its	26
court	26
live	26
your	26

when	26
nominatio	25
people	25
some	25
delegate	25
business	25
solar	25
primary	25
over	24
party	24
no	24



CRAWL 2:

Seed 2: http://www.apple.com/mac/

Max page: 100

Domain Restriction: None

Top 100 words:

Terms	Frequency
and	2318
the	1642
to	1575
your	1131
you	1116
with	970

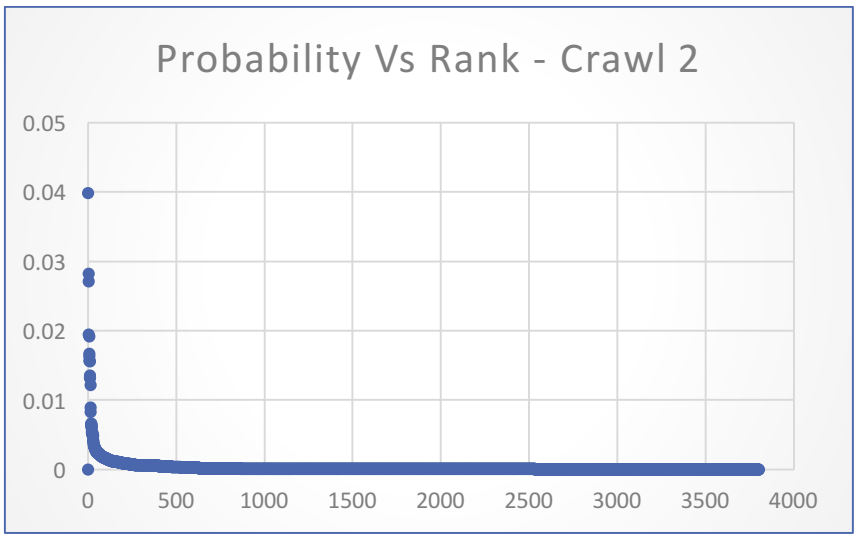
a	948
for	912
more	905
in	790
learn	772
of	765
mac	707

apple	520
on	480
or	386
pro	379
can	372
that	361
work	357

up	321
all	305
is	303
from	300
it	289
	287
store	282
apps	246
create	236
an	235
devices	211
new	206
use	199
get	189
at	189
we	188
its	185
like	182
about	177
business	174
icloud	168
power	168
as	163
x	161
by	156
powerful	150
macbook	149

music	146
seamlessly	145
final	143
have	143
cut	142
do	141
ios	139
2	138
are	137
1	136
features	135
built	135
any	131
everything	129
share	127
need	127
available	126
iphone	126
now	126
beautiful	126
than	124
video	122
before	119
find	118
shop	114
free	114
3	114

ipad	112
make	111
every	110
library	110
never	108
our	106
photos	106
help	104
support	103
people	102
be	102
5	102
most	101
core	101
just	100
things	100
who	100
easy	99
performan	98
even	98
tools	98
motion	97
br	97
editing	95
processor	95
logic	95



CRAWL 3:

Seed 3: http://www.nytimes.com/

Max page: 100

Domain Restriction: None

Top 100 words:

Terms	Frequenc
the	7065
a	3313
to	3193
and	2881
of	2753
in	2514
by	2158
	2030
for	1270
that	1023
is	940
on	919
with	749
he	679
new	664

was	628
at	625
it	584
said	582
as	581
mr	553
his	528
from	515
but	473
are	469
has	419
i	413
who	405
an	400
have	396
times	395

2016	390
more	388
she	382
not	382
york	378
this	373
be	373
its	353
her	349
they	345
story	338
reading	313
main	311
continue	308
had	303
about	299

trump	291
or	289
one	277
advertisem	275
you	266
what	263
up	260
will	256
out	248

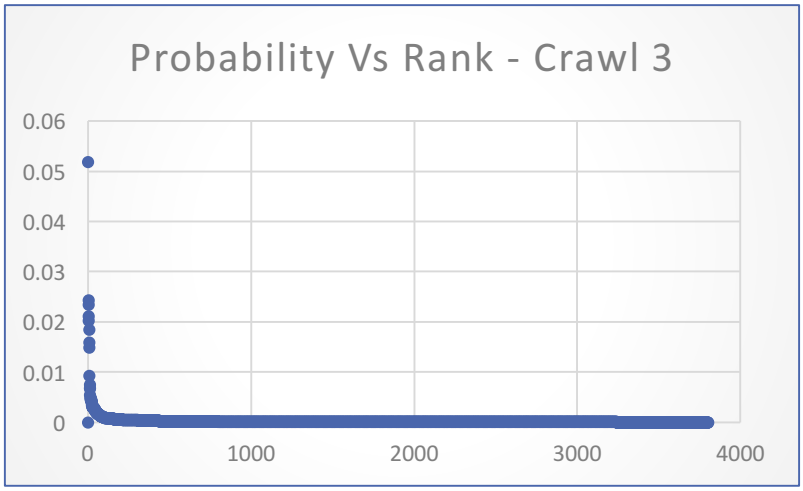
no	242
their	238
were	237
all	233
page	229
how	221
april	221
s	220
which	217

when	212
been	212
de	211
first	204
like	200
after	200
would	195
than	190
your	183

follow	175
1	175
some	172
skip	170
donald	169
years	168
if	167
percent	162
can	160

27	157
there	154
two	154
la	154
could	152
we	150
results	145
into	145
other	143

year	141
9	138
republican	134
where	131
even	131
section	130
state	128
last	128



REFERENCES

PAPERS:

[1]. M. Kovacevic, M. Diligenti, M. Gori, M. Maggini, and V. Milutinovic, —Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification,‖ In the Proceedings of 2002 IEEE International Conference on Data Mining (IC DM'02), Maebashi City, Japan, December 2002.

[2]. S.-H. Lin and J.-M. Ho, —Discovering Informative Content Blocks from Web Documents,‖ In the Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD'02), 2002.

[3]. L. Yi, B. Liu, and X. Li, —Eliminating Noisy Information in Web Pages for Data Mining,‖ In Proceeding of the 9th ACM SIGKDD International Conference, 2003.

[4]. T. Win and K.N.N. Tun, —Noise Elimination for Improving Web Information Extraction,‖ In the Proceedings of the Seventh International Conference on Computer Applications, 2009.

[5]. Tim Weninger, William H. Hsu, Jiawei Han --- CETR - Content Extraction via Tag Ratios --- WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA. ACM 978-1-60558-799-8/10/04.

[6]. S. Gupta, G. E. Kaiser, and S. J. Stolfo. Extracting context to improve accuracy for html content extraction. In WWW (Special interest tracks and posters), pages 1114–1115. ACM, 2005.

Online references:

http://www.programcreek.com/2012/12/how-to-make-a-web-crawler-using-java/
http://www.programcreek.com/2009/11/how-to-write-a-java-crawler/
http://stackoverflow.com/questions/19332982/parsing-robot-txt-using-java-and-identify-whether-an-url-is-allowed
http://www.java-tips.org/java-se-tips-100019/15-javax-swing/1393-how-to-create-a-search-crawler-in-java.html
https://github.com/datumbox/datumbox-framework
https://github.com/FeiSun/ContentExtraction/blob/master/ContentExtraction/textdensity.cpp
http://ramb.ethz.ch/CDstore/www2010/www/p971.pdf
http://users.dsic.upv.es/~jsilva/CNR/#Experiments