MET CS688 C1

WEB ANALYTICS AND MINING

ZLATKO VASILKOSKI

WEB CRAWLING

CS688 Exam Sample Type of Questions

1. Multiple Choice/Response Type of Question

Q3: Which of the following statements is NOT correct? The web mining technology applies to mining data in a form of:

Choices:

a) Emails
b) Web pages
c) RSS feeds
d) Databases such as genome databases
e) Plain text
f) Only the information in a textual format, excluding audio or video.

Answer: f)

CS688 Exam Sample Type of Questions

2. Paragraph Style Type of Question

The classic Reuters-21578 collection which is a collection of 21,578 newswire articles is a benchmark for text classification evaluation. You can find two sets of samples of it in your “tm” library ("~/tm/texts/" folder). Consider the "crude" dataset from the “tm” library, and show the R code you would write to implement the following tasks:

Tasks (show your code):

a) Create an R object containing the dataset location. Make sure you don't "hard code" the path so your code can be used on any other computer.
b) Obtain the Corpus of the "crude" dataset. How would you save it? In what format and in how many files?
c) Find the number of documents in the Corpus?
d) How would you examine the content of the third document?
e) How would you find the most frequent terms in the document term matrix?
f) How would you subset the corpus by a query containing the key words: "prices", "crude", "oil"?
g) How would you inspect the first 6 entries in your subset?

Hint: You can refer to Module Examples regarding these questions. For example you can subset the corpus "reuters" by keyword such as “TEXACO” in the corpus field "heading" (which is a sub-field in "meta") by:
grep("TEXACO", meta(reuters, 'heading'))
This kind of hint will not be included on the final exam!

Web Mining and Analytics, Content

- Web Crawlers
- Indexing
- A Web Crawling Illustration
- Understanding Search Performance
- Using Shiny
- Ranking

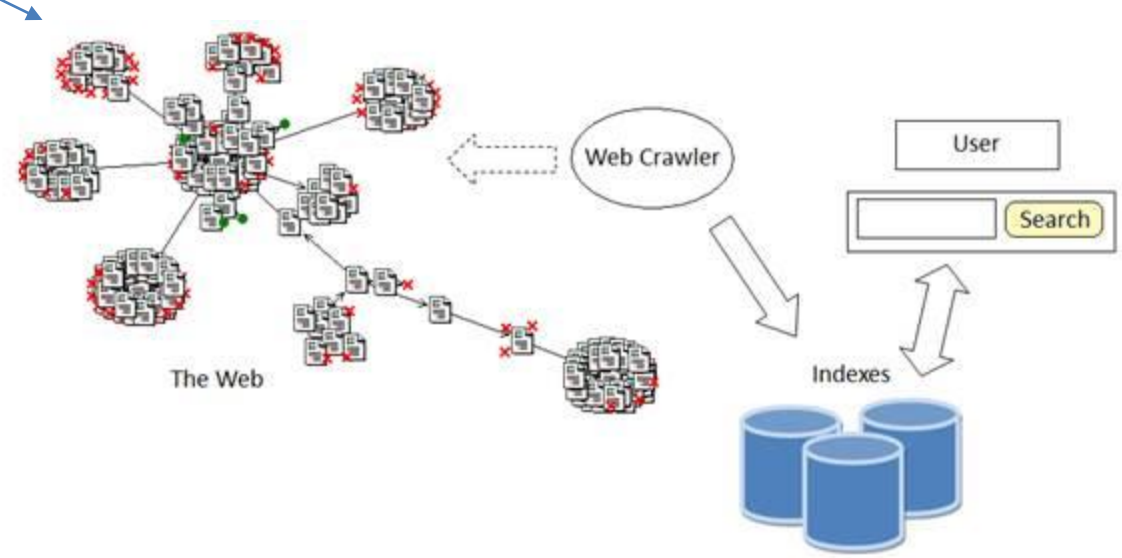
Web Mining

- Similar techniques to text mining
  - Difference, in the use of a search engine (the information data is on the web)
- Gathering pages from the web and indexing them in order to support a search engine.
- Web mining technology applies to mining data in a variety of form such as:
  - Web pages
  - A collection of SGML (Standard Generalized Markup Language) generalized markup language for documents
  - XML (Extensible Markup Language) documents, textual data format intended to be both human and machine readable.
  - Genome databases (for example GenBank, PIR)
  - Online dictionary (for example Oxford English Dictionary)
  - Emails or plain texts on a file system.

Components of a web search engine

- The techniques for web mining are similar to the ones used for text mining with the exception of the use of a search engine.

- The search engine has the following architecture
  - Content Aggregator (Crawling Subsystem, Google, Yahoo etc. search)
  - Indexing Subsystem
  - Search Interface
  - User (Content Consumer)



Web Crawlers

- A web crawler fetches, analyses and files information from web servers.
- Web crawlers (sometimes referred to as a spider) can copy all the indexed pages they visit for quicker processing by a search engine.

- The basic operational steps of a hypertext crawler are
  - Begin with one or more URLs that constitute a seed set
  - Fetch the web page from the seed set
  - Parse the fetched web page to extract the text and the links
    - Extracted text is fed to a text indexer
    - Extracted links (URLs) are added to URLs whose corresponding pages have yet to be fetched by the crawler
  - The visited URLs are deleted from the seed set

- It is a recursive traversal of a web graph where each node is a URL.
- Multi-threaded design to process a large number of web pages quickly (a fetch rate).
  - A fetch rate of hundred pages each second will fetch a billion pages in 1 month long crawl.
  - This is a small fraction of the static Web at present (massively distributed parallel computing typically used).

Search engine indexing

- Web (Internet) indexing refers to various methods for indexing the contents of a website or of the Internet as a whole.
- Metadata web indexing - assigning keywords or phrases to web pages or web sites within a metadata tag field
  - So that the web page or web site can be retrieved with a search engine that is customized to the search in the keywords field.
- Web pages with frequent changes would require dynamic indexing.
- For very large data collections like the web, indexing has to be distributed over computer clusters with hundreds or thousands of machines.
- The basic steps in constructing an index - A term–document ID pairs as described in Module 3.
- Considering the large data collections, index implementations consists of
  - Partitioning by **terms**, also known as **global** index organization.
  - Partitioning by **documents** (more common), also known as **local** index organization.

WebSPHINX Crawler

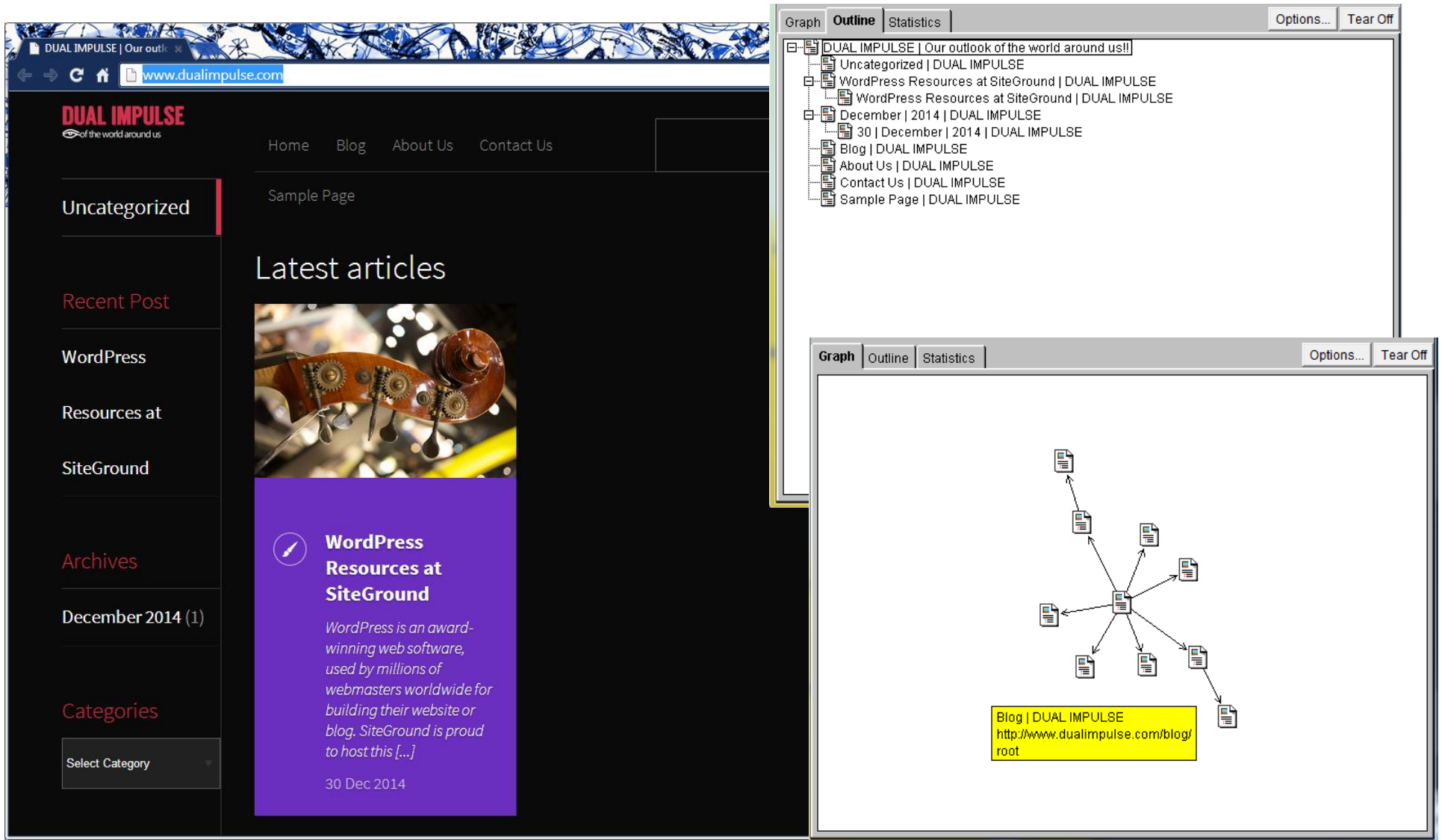
- WebSPHINX is a GUI tool and a Java class library that can be downloaded from:
  http://www.cs.cmu.edu/~rcm/websphinx/#download

- WebSPHINX – a web crawler that allows you to experiment with some basic crawls automatically, over a small part of the web (a single web site).
  - Open source
  - Intended for a personal use (not a professional)
  - Customizable
  - Visualizes the collection of the visited web pages as a graph.
  - It saves the visited web pages for offline browsing.
  - It also can extract pattern of text matching criteria from the collection of pages.

- **Note**: Make sure you crawl over the web sites that would not restrict your access after the first test crawl. Many web sites will restrict your access after crawling with WebSPHINX.

Try to maybe crawl over your classmate's web site from the Google Analytics Project.

- Here is crawl of the website http://www.dualimpulse.com/

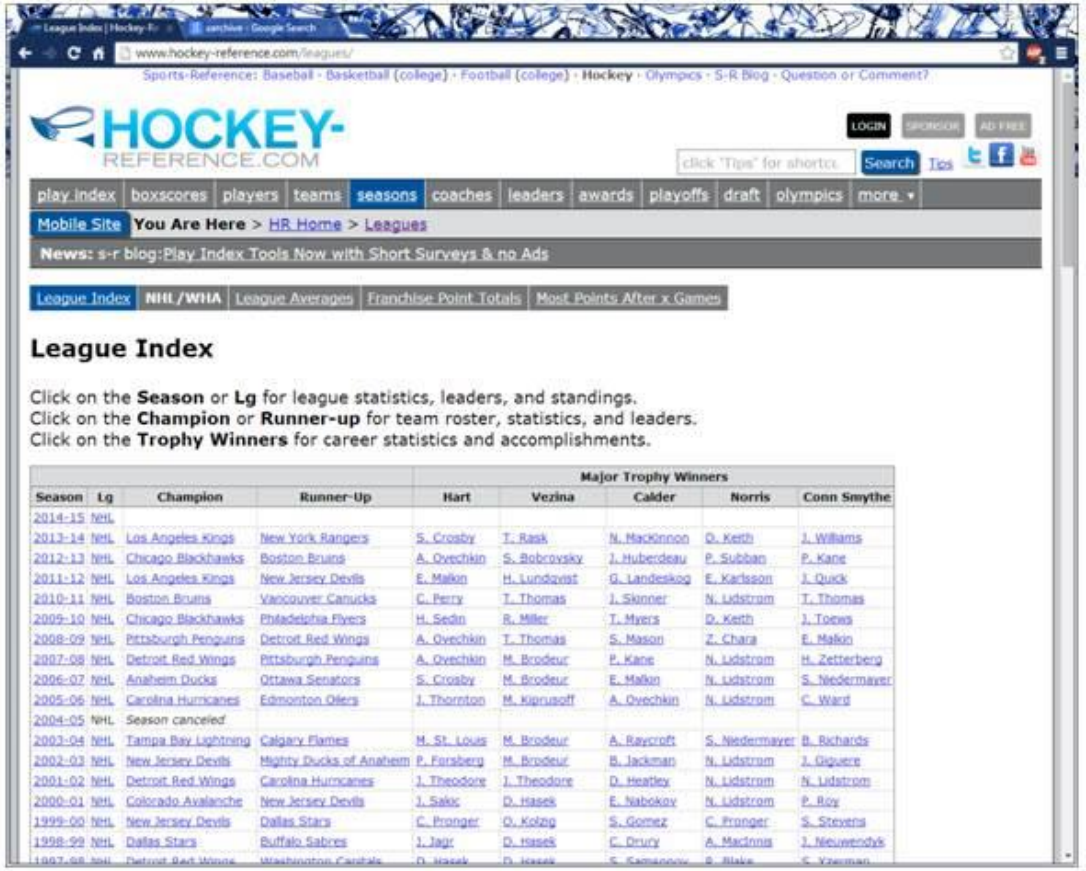


WebSPHINX Illustration

The following URL "http://www.hockey-reference.com/leagues/"

contains the hockey league reference information on the NHL seasons going back to 1917.

This site has a lot of interlinked web pages containing relevant and overlapping information, thus making it an illustrative test example.

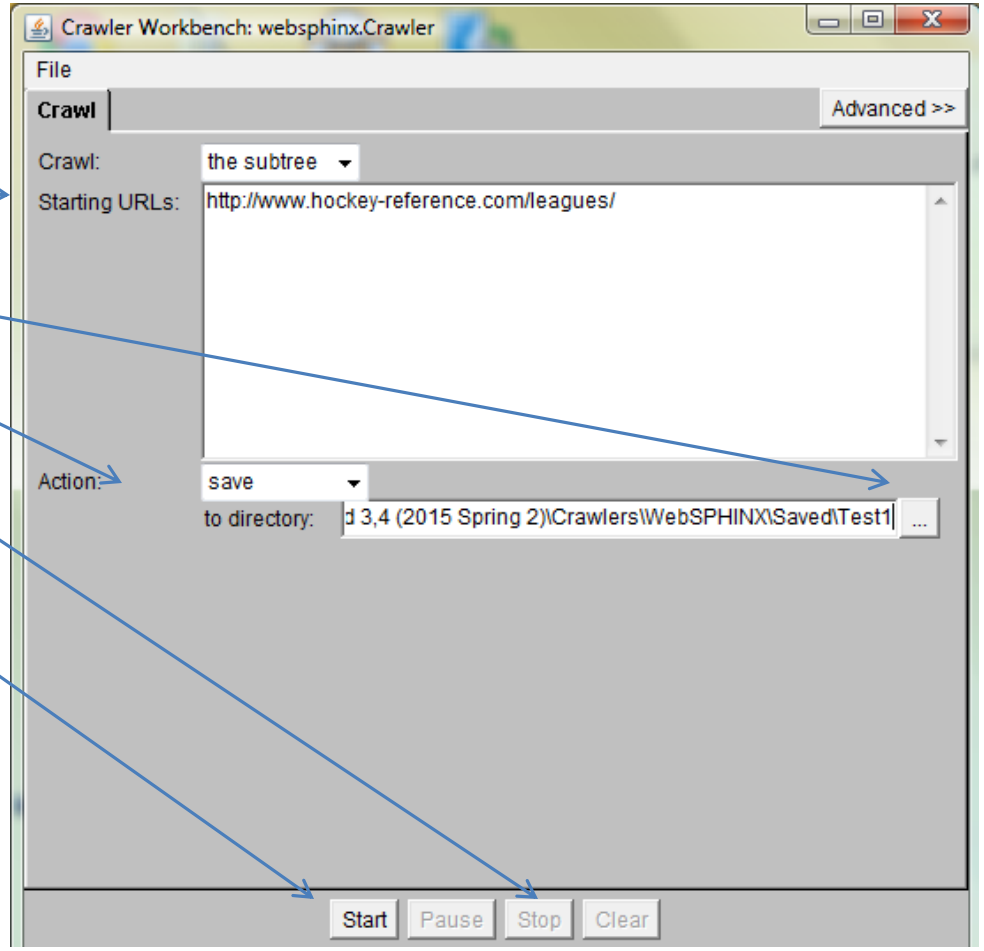


Running WebSPHINX

Choosing the crawl options for WebSPHINX.

- Type the URL
- Specify Action - save
- Enter location to save visited HTML files
- Then click start

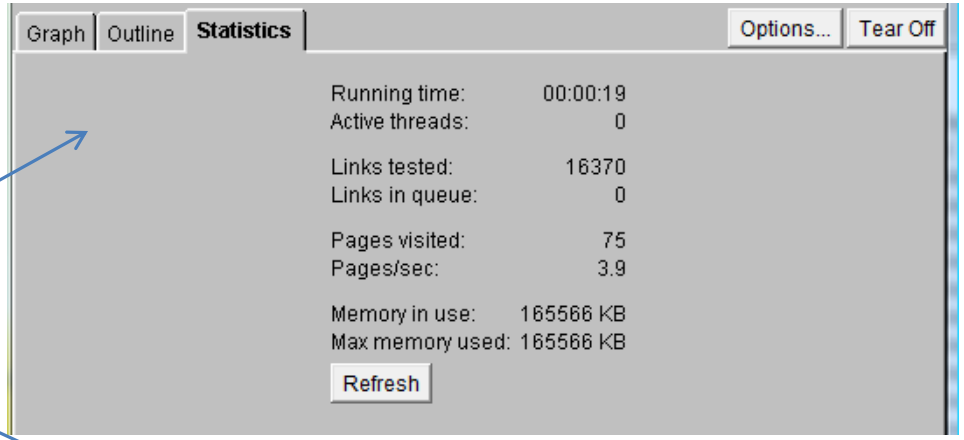
After some time you can stop the crawl

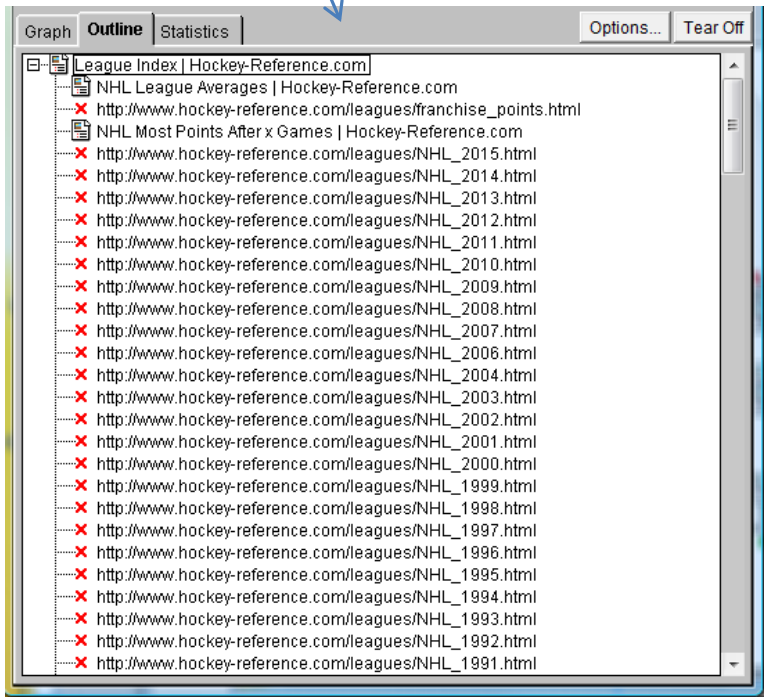


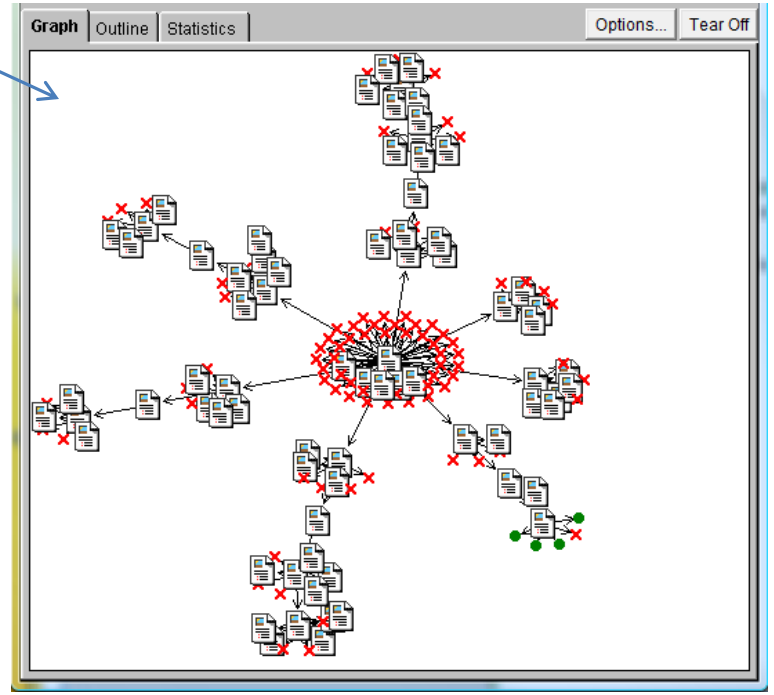
Running WebSPHINX

The web pages explored are visualized

- As a graph
- As a list
- The statistics about the crawl







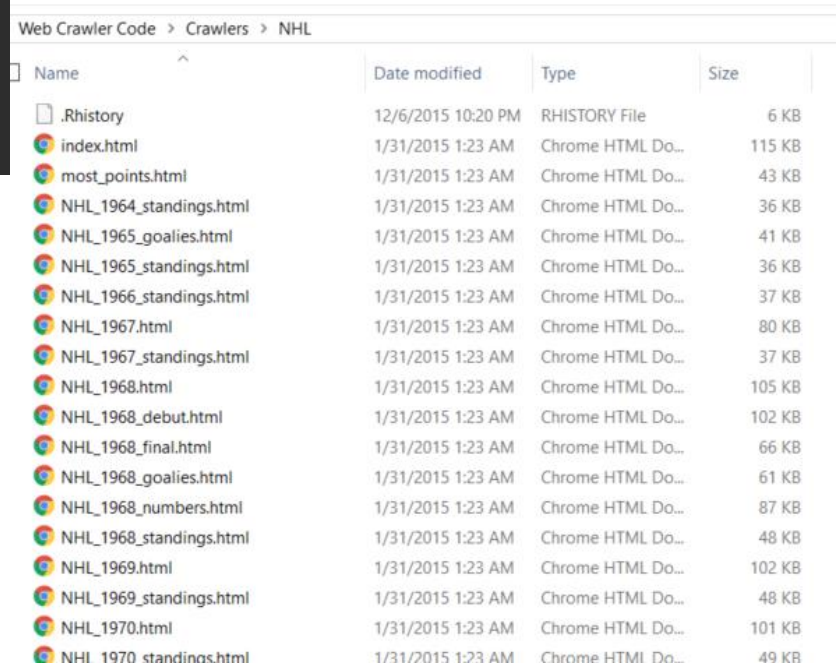
Analyzing WebSPHINX results with R

The following code illustrates how the saved data from WebSPHINX crawl can be analyzed with R.

```r
# Module 4 Code

### --- Example 1: Analyze WebSPHINX results. --------
library(XML)
dir <-file.path(paste0('.\\Crawlers\\','NHL\\'))
HTML.dataset <- list.files(dir,pattern ="html") # List of all saved HTML files @ location "dir"

# Function to strip the table data from the HTML files
sieve.HTML <- function(URL) {
  table <- readHTMLTable(URL) # Read HTML table into a list
}

temp.HTML.text <- lapply(as.list( paste0(dir,HTML.dataset)),function(x) sieve.HTML(x)) # Get all the text from the saved HTMLs

query <- "Boston Bruins"
temp <- grep(query, temp.HTML.text[[1]][[1]]$Champion)
# [1]  5 51 53 82 84 94
temp.HTML.text[[1]][[1]]$Season[temp]

```

This code gives the seasons when "Boston Bruins" were champions.

Saved data from a crawl over the hockey league website resides in a folder Crawlers/NHL/



Lab Project: WebSPHINX Crawler

- Use WebSPHINX to crawl over one of the web sites (yours or one of your classmate's) created for the Google Analytics project.
- **Note**: Make sure you crawl over the web sites that would not restrict your access after the first test crawl. Many web sites will restrict your access after crawling with WebSPHINX.
- Present your crawl results in a slide similar to the illustrations on the previous slides.
- Make sure to include the URL and the screenshots of the web pages as a graph.
- Submit your slide on Blackboard