Development of an Intelligent Web Based Dynamic News Aggregator Integrating Infospider and Incremental Web Crawling Technology

Ihekweaba Chukwugoziem, Ubochi Chibueze Nwamouh, Okereke Eze Aru

Corresponding Author: chibuezeubochi@gmail.com

Abstract- The World Wide Web is a rapidly growing and changing information source. This reality is gradually replacing the traditional way users obtain news or information. Traditionally, individuals get their news or information from print media, such as newspapers and magazines. Although, the advent of the internet has made things a lot easier by making this digitalized news accessible from anywhere in the world, either through news websites or dedicated applications. However, the growth and change rates make the task of finding relevant and recent information harder. Users are still faced with the challenges of visiting numerous websites just to get updated or informed on a specific type of news. This creates a problem as users have to always memorize different URLs and visit numerous websites just to view a specific type of news. Therefore, the need to develop an intelligent web based dynamic news aggregator that will provide a digital platform for individuals to easily find news pertaining to a particular topic in real time becomes imperative. It crawls the web, searches for news agencies and return a specific news of interest to the user. To address the shortcomings of existing news aggregators, this work was implemented by integrating the intelligent web based dynamic news aggregator, into an infospider web crawling technology. This is achieved applying a stochastic selector and incremental web crawling technology that crawls the entire seed urls. This system was implemented with the PHP scripting language developed to access the PHP-crawler using Aptana Studio as the Integrated Development Environment (IDE), Bootsrap3 and jQuery were used to provide a set of style sheets and JavaScript libraries to simplify the client-side scripting. The application was deployed and tested using the apache web server and a personal computer. The results are presented and discussed.

Index Terms: News aggregator, web crawler, url frontier, seed url, crawling, Infospider and Incremental

1 INTRODUCTION

In computing, an aggregator is a client software or a web application which aggregates syndicated web contents in one location for easy viewing. Basically, the news aggregator uses the extensible markup language to structure pieces of information to be aggregated and displays the information in a user-friendly interface. Of a truth, the World Wide Web is a rapidly growing and changing information source. Its growth and change rates make the task of finding relevant and recent information harder. Therefore, developing an intelligent news aggregator integrating info-spider and incremental web crawling technology will help gather and distribute content more effectively after it has been appropriately organized and processed with respect to customers' requirements. It will saliently control and collect information according to clients' criteria as opposed to the Print media and numerous websites that need to be visited for specific types of news.

Traditionally, individuals get their news or information from the Print Media such as Newspapers and magazines. Although, the advent of the internet has made things a lot easier as most of these contents have been digitized and viewed from anywhere in the world either through a dedicated application or a website. Even upon this digital improvement, users still face certain issues such as visits to numerous websites in a bid to view a specific type of news. For example, someone interested in politics would have to visit punch.ng, Vanguardngr.com, Sunnewsonline.com and other news related websites just to view news on just politics.

This creates a problem as users have to always memorize different URLs and also visit numerous website just to view this specific type of news; politics.

To solve the problem enumerated above, this work is intended to create an intelligent dynamic content aggregator, integrating Infospiders and the incremental web crawling technology, which would house different categories of news, from different news websites, by properly ordering and grouping them in their specific categories, and to be subsequently served the users on one platform.

Existing news aggregators take up different realities with wild range of differences, differences that most times are based on functions; such as the type of syndicated web pages they aggregate and their crawling methods. Many researchers have developed news aggregators that carry out unique functions. Though some of these aggregators are "static" with external links, crawling is not implemented, the news are manually typed into the news aggregators, others are uniquely integrated with web crawling technology with same or different crawling algorithms.

The web crawler of a news aggregator, also known as harvester, spider, or robot, is an application that browses the World Wide Web and automatically downloads web pages. The basic operations of a hypertext crawler is indicated in figure1.

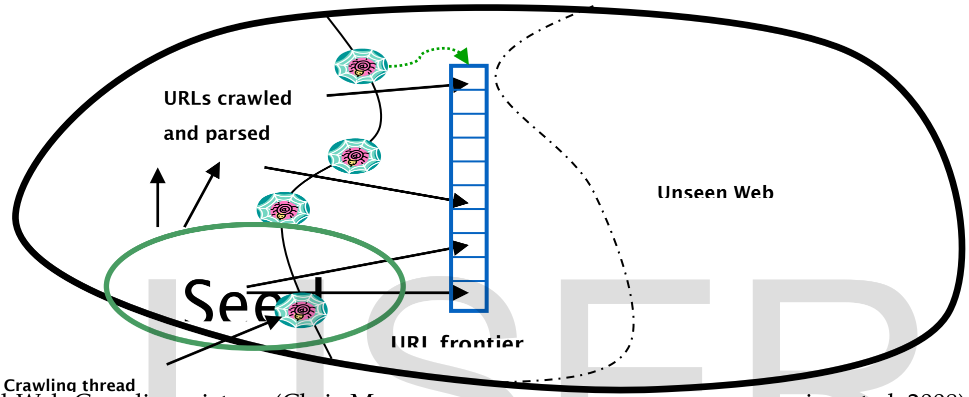


Figure1: Updated Web Crawling picture (Chris Manning et al, 2008)

Here, the fetched page is parsed, to extract both the text and the links from the page. The extracted links URLs are then added to a URL frontier, which at all times consists of URLs whose corresponding pages have yet to be fetched by the crawler. Initially, the URL frontier contains the seed set, as pages are fetched, the corresponding URLs are deleted from the URL frontier. The entire process may be viewed as traversing the web graph.

The crawling methodology of the new system differs in its crawling algorithms but stills maintains the basic architecture of a web crawler as expressed by Christopher D. (2008). This architecture is shown in figure2 below.

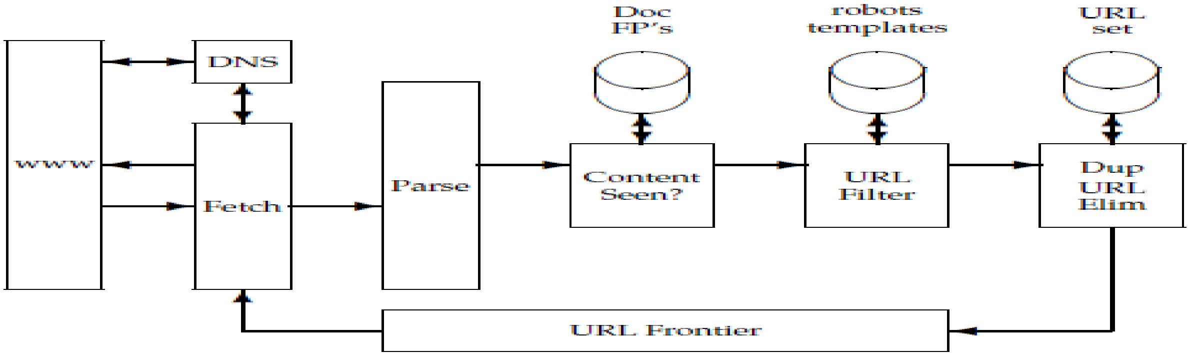


Figure 2: Basic Architecture of a Web Crawler (Christopher D. 2008)

The seemingly simple recursive traversal of the web graph is complicated by the many demands on a practical web crawling system as shown in figure 2. The simple scheme outlined above for crawling demands several modules that fit together as shown in Figure 2 above. The URL frontier, contains urls yet to be fetched in the current crawl, in the case of continuous crawling, a url may have been fetched previously but is back in the frontier for re-fetching. The Domain Name Server (*DNS) resolution* module determines the web server from whichto fetch the page specified by a Uniform Resource Locator (URL). The fetch module uses the http protocol to retrieve the web page at a url. The parsing module extracts the text and set of links from a fetched webpage. The duplicate elimination module determines whether an extracted link is already in the URL frontier or has recently been fetched. The crawler begins with one or more URLs that constitute a *seed set*. It picks a URL from this seed set, then fetches the web page at that URL. The text and the links are then extracted from the page. The extracted links are then added to a *URL frontier*. The basic architecture of a Web Crawler must be distributed, scalable, efficient, polite, robust, and extensible while fetching pages of high quality. Therefore, the new system must be uniquely different by examining the effects of each of the components in figure2 above, and then tries to get them feasible by integrating this reality into an aggregating content. In this light, the new system considers the crawling algorithms (Infospiders and incremental web crawling algorithms) that will be integrated to follow the design of the Mercator crawler (shown in figure3) that has formed the basis of a number of research and commercial crawlers, especially in the area of web crawling.

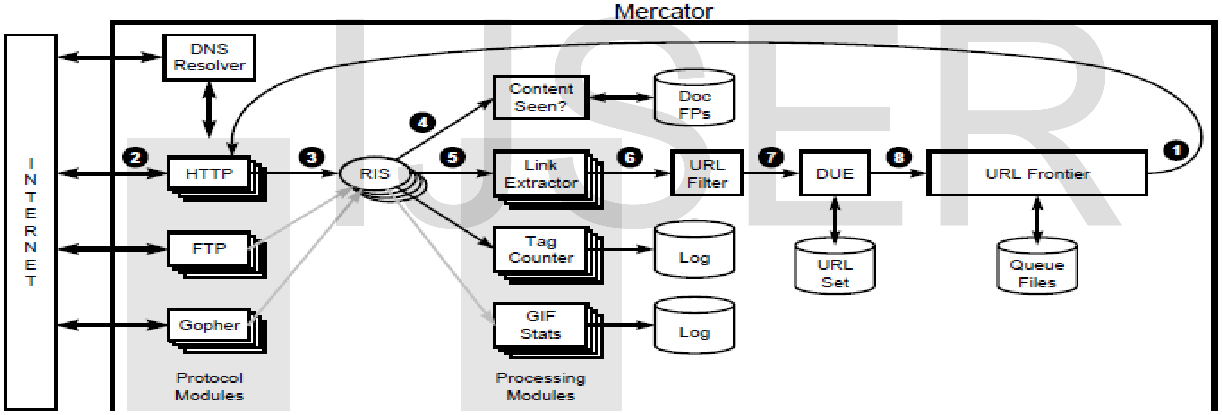


Figure3: Mercator's main components. (Marc Najork and Allan Heydon, 2001)

The basic operation executed by any web crawler takes a list of seed URLs as its input and repeatedly executes the following steps; Remove a URL from the URL list, determine the IP address of it's host name, download the corresponding document, and extract any links contained in it. For each of the extracted links, ensure that it is an absolute URL (derelativizing it if necessary), and add it to the list of URLs to download, provided it has not been encountered before. If desired, process the downloaded document in other ways. All web crawlers associated with news aggregators reflect same operation as described by Marc and Allan but are different in their crawling technology. Existing crawling technology include: PageRank crawling technology, Naive Best-First crawling technology, SharkSearch Crawling technology, Focused Crawling technology, and Context Focused crawling technology.

Existing news aggregators such as Google news aggregator, uses an algorithmic process of its web crawler known as Googlebot and the PageRank crawling technology to determine how often sites should be crawled, thereby keeping the news aggregator up-to-date.

The PageRank of a page A:

$$\textbf{\textit{Page Rank of A}} = \frac{\textbf{\textit{Previous PR of the webpage pointing to A}}}{\textbf{\textit{Number of outgoing links of that webpage pointing to node A}}}$$

If we consider six web pages A, B, C, D, E, F each is pointing to the others as shown in figure4 below.

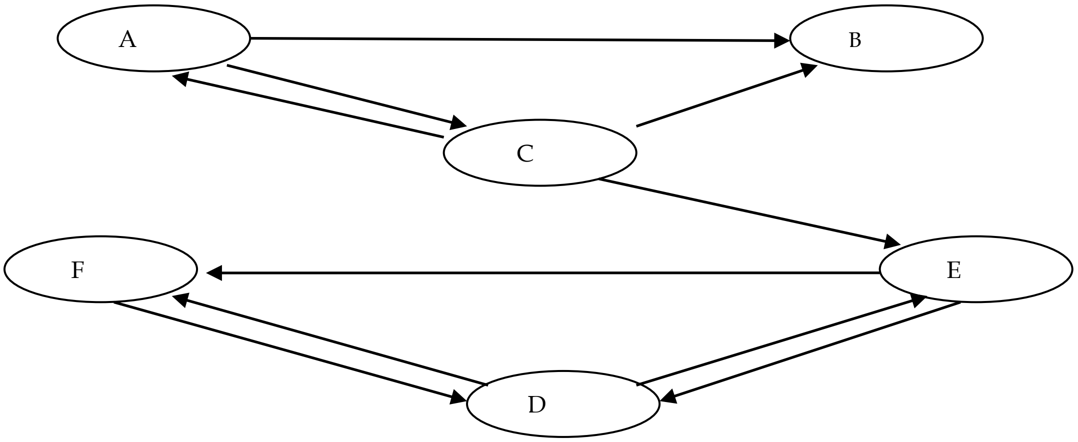


Figure4: Six pages, each pointing to each other

Table1: Result of PageRank of figure4

Pages	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/6	1/18	1/36	2
B	1/6	5/36	1/18	3
C	1/6	1/12	1/36	2
D	1/6	¼	17/72	6
E	1/6	5/36	11/72	4
F	1/6	1/6	14/72	5

1. Iteration 0: at iteration 0, all the web pages have the same Page Rank. It is $\frac{1}{\text{total number of pages in the website}}$. Since figure4 has six webpages, the PageRank at iteration0 for all webpages is $\frac{1}{6}$.

2. Iteration 1: let's take for instance webpage A, recall, the Page Rank of page A is:

$$PageRank\ of\ A = \frac{Previous\ PR\ of\ the\ webpage\ pointing\ to\ A}{Number\ of\ outgoing\ links\ of\ that\ webpage\ pointing\ to\ node\ A}$$

Therefore, Page Rank of A = $\frac{\frac{1}{6}}{3} = \frac{1}{18}$ **Page Rank of B =** $\frac{\frac{1}{6}}{2} + \frac{\frac{1}{6}}{3} = \frac{5}{36}$

Page Rank of C = $\frac{\frac{1}{6}}{2} = \frac{1}{12}$

Page Rank of D = $\frac{1}{6} + \frac{\frac{1}{6}}{2} = \frac{1}{4}$

Page Rank of E = $\frac{\frac{1}{6}}{2} + \frac{\frac{1}{6}}{3} = \frac{5}{36}$

And Page Rank of F = $\frac{\frac{1}{6}}{2} + \frac{\frac{1}{6}}{2} = \frac{1}{6}$

3. Iteration 2: at this stage the previous PageRank becomes iteration1.

Recall:

$$PageRank\ of\ A = \frac{Previous\ PR\ of\ the\ webpage\ pointing\ to\ A}{Number\ of\ outgoing\ links\ of\ that\ webpage\ pointing\ to\ node\ A}$$

Therefore, Page Rank of A = $\frac{\frac{1}{12}}{3} = \frac{1}{36}$

Page Rank of B = $\frac{\frac{1}{18}}{2} + \frac{\frac{1}{12}}{3} = \frac{1}{18}$

Page Rank of C = $\frac{\frac{1}{18}}{2} = \frac{1}{36}$

Page Rank of D = $\frac{1}{6} + \frac{\frac{5}{36}}{2} = \frac{17}{72}$

Page Rank of E = $\frac{\frac{1}{12}}{3} + \frac{\frac{1}{4}}{2} = \frac{11}{72}$

Page Rank of F = $\frac{\frac{5}{36}}{2} + \frac{\frac{1}{4}}{2} = \frac{14}{72}$

According to the iterative algorithm of the PageRank crawling algorithm, at iteration2 the PageRank of the websites can be calculated. This is because further iteration will still yield the same PageRank result. Therefore, time wastage and redundancy should be avoided.

4. At the end of the desired iteration, the webpage with the highest number will have the highest page rank, and so on. In table1, page D has the highest PageRank, followed by page F, page E, page B, page A and C

New pages have less page rank and they take much time to be listed and gain high ranks.

According to Rinki Tyagi (2016), the limitation of the PageRank crawling technology is that the PageRank scores do not reflect current events and are not calculated at the time of search – this would be too expensive and slow. This means that a recently updated page is not determined to be an authority on a particular topic until after it has gained exposure as well as paths from other authority pages. PageRank algorithm is less relevant to the query of user as it ignores that the page is relevant or not.

Other existing news aggregators such as Drudge news, HuffPost, Fark, Zero Hedge, The Daily Beast, World News, and Newsvine aggregates content by copying and pasting of links, editing of aggregated contents and articles by users which are "seed" links to external content.

Despite the unique reality and functionality of the crawling algorithm used by the existing systems, the new system moves a step further to manifest a non-comparable but unique news aggregator with similar functionality with the Google News, Yahoo News, Drudge Report, Huffington Post, Fark, Zero Hedge, Newslookup, The Daily Beast, World News (WN) Network and Newsvine news aggregator in aggregating syndicated web pages (just as other news aggregators). This is achieved by employing a different crawling algorithm that involves the amalgamation of two unique crawling algorithms. The proposed system does not condemn the crawling algorithm being used by the existing news aggregators but tries to implement a reality that would make a difference and add to the body of knowledge by showcasing that the combination of two crawling algorithms (Incremental and Infospiders) can do better in a web crawling exercise.

2 SYSTEM DESCRIPTION

Figure5 below is an illustration of the system implemented. The complex design activity of aggregating news via incremental and Infospiders web crawling was divided into several small sub activities as explained subsequently using logical designs which coordinate with each other.

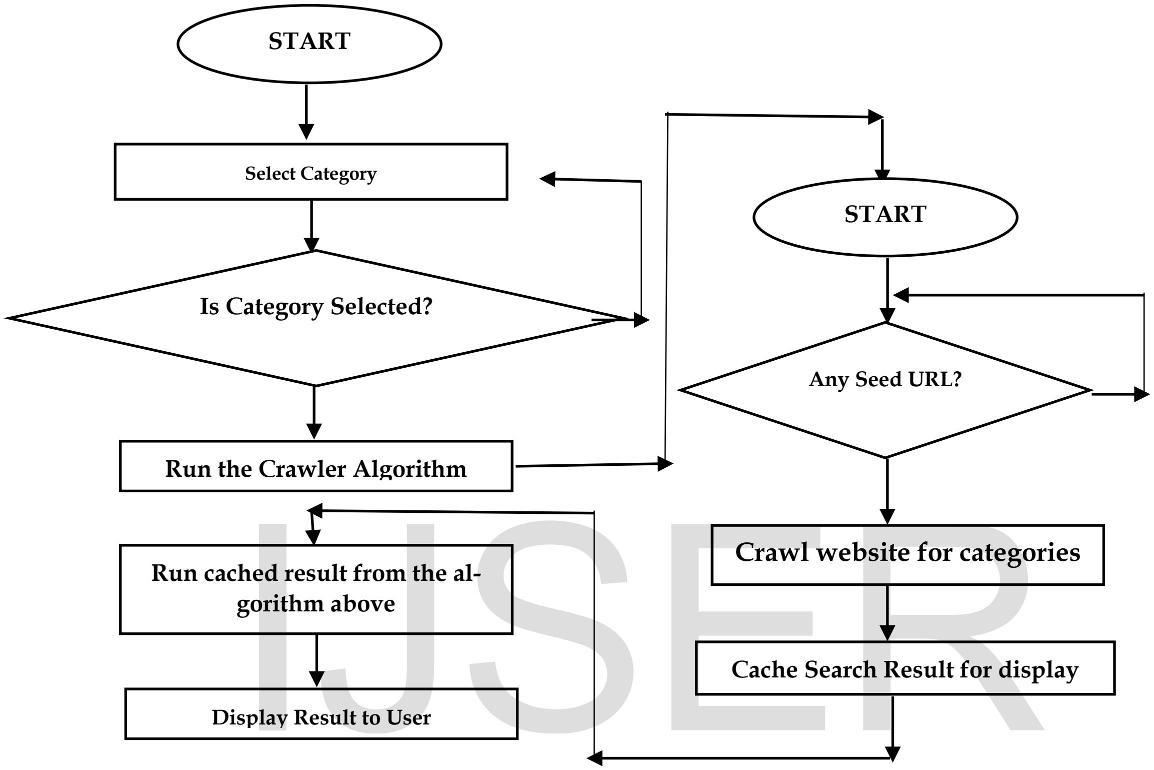

START
Select Category
Is Category Selected?
Run the Crawler Algorithm
Run cached result from the algorithm above
Display Result to User
START
Any Seed URL?
Crawl website for categories
Cache Search Result for display

Figure5: New system flow chart

2.1 Initializing The URL Frontier

The URL frontier is the data structure that contains all the URLs that need to be downloaded. The URL frontier was initialized with SEED URLs:

QUEUE1 = https://www.Vanguardngr.com/ which is the URL for Vanguard newspaper

QUEUE2 = https://www.Sunnewsonline.com which is the URL for Sun newspaper

QUEUE2 = theNationonlineng.net/ which is the URL for Nation newspaper, the URL frontier implementation is shown in figure5.

The web crawler works by performing a breath-first traversal of the web, starting from the pages in the seed set (Seed URLs), which was implemented using a FIFO (First In First Out) queue. Meaning that elements are dequeued in the order they were enqueued.

2.2 Frontier Implementation Details

Since, it is considered socially unacceptable to have so many HTTP requests pending to the same server. If multiple requests are to be made in parallel, the queue's *remove* operation should not simply return the head of the queue, but rather a URL close to the head whose host has no outstanding request. In observing this reality, the following web crawler was implemented:

- Polite: respects both implicit and explicit politeness considerations. Respect explicit consideration is by respecting the ROBOT.txt, do not crawl pages that are disallowed, while Respect implicit consideration is by not bombarding a server too often
- Robust: immune to spider trap and other malicious behaviour from web servers.
- Capable of distributed operation: designed to run on multiple distributed machines.
- Scalable: designed to increase crawl rate by adding more machines
- Performance/Efficiency: able to avoid processing duplicate pages thereby avoiding waste of processing power.
- Fetch pages of high quality first: able to give more priority to pages with high relevance
- Continuous operation: continue to fetch copies of a previously fetched page.

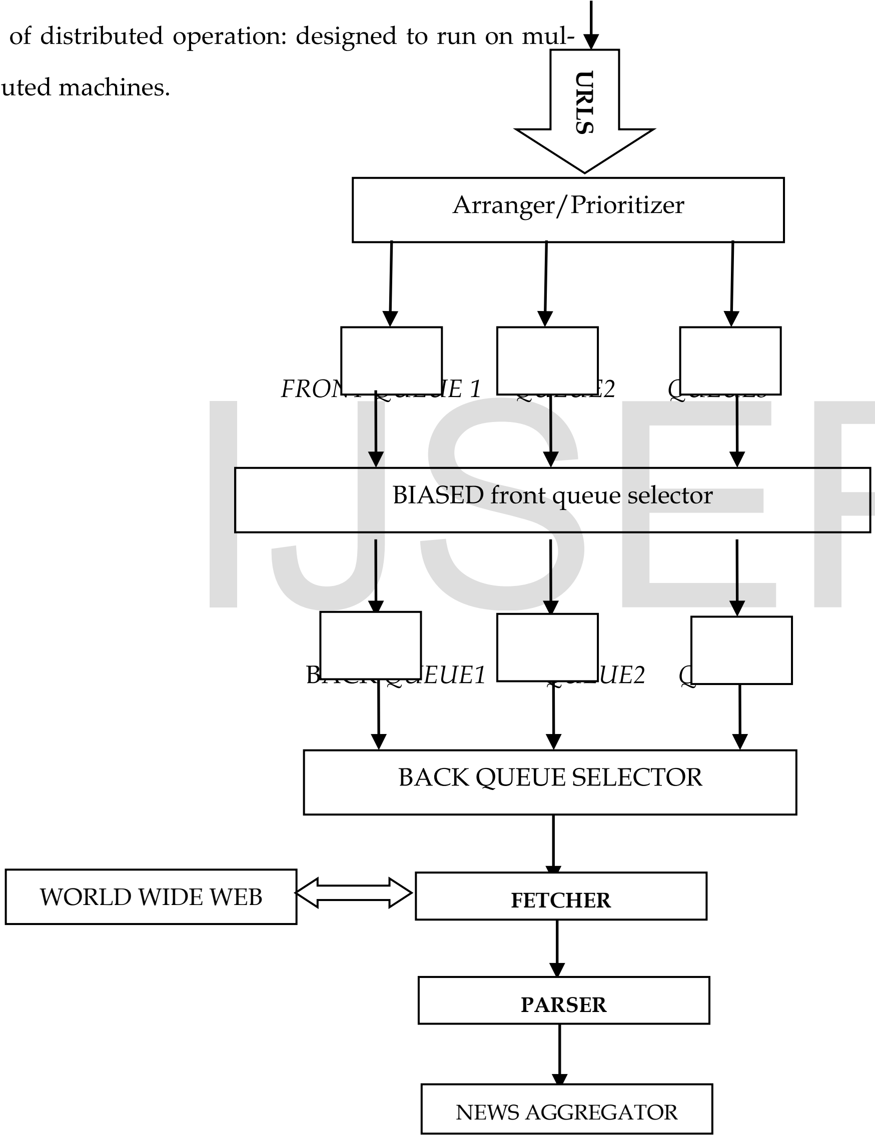


Figure6: Frontier Implementation.

Here, the arranger or priotizer pulls seed URLs into the front queues based on first-in-first-out (FIFO). Each front queue corresponds to one particular server. The BIASED front queue selector selects the URLs into the back queue considering traffic on that URL, so there is every possibility that the first seed URL at the front queue may likely not be the first at the back queue.

On subsequent crawling, fetched URLs are queued under their respective seed URLs before they will be parsed. A threshold time stamp of 30seconds was assigned to the back queue. Subsequent queues are twice the threshold timestamp of the previous queue, thereby making the first queue of the back queue to have the least timestamp so as to limit the number of outstanding HTTP requests to any given web server. The threshold time stamp is the value of time before a web crawler is not allowed to hit the web server for a crawl. This is done so that our system respects the implicit consideration of politeness which all web crawlers must abide by. That is “do not bombard a server too often”.

Though crawling is a continuous action, it is salient to implement a reality that signals our system when to recrawl a page after it has been crawled. Recrawling a page when there is no update is a waste of time. If no update, then no recrawl. For every update then recrawl. This was achieved by making the URL frontier an intelligent agent using Infospiders and Incremental web crawling technology in the system implementation.

The system incrementally refreshes the existing collection of web pages based on the next threshold timestamp. Thereby, resolving the problem of the freshness of the pages. The Infospiders representation consisted of categories and a feedforward neural net. The system uses the feedforward neural network to adapt to the environment and makes a recrawl after a cycle threshold timestamp, if and only if there is any update on that webpage. The following News categories were implemented in this system: “Politics”, “Entertainment”, “Sport” and “Business”. This is based on the intuition that people are usually interested in News amongst this set.

The news aggregator was designed to be the systems output. After crawling, the fetcher takes the headlines and their corresponding ULRs and outputs on the aggregator. These headlines are arranged under each authoritative news site and are accessed by selecting a category of choice as shown in figure7. For each of the categories selected, the corresponding outputs appear.

3 MATERIALS AND MEHTOD

3.1 Materials

The following materials were used to implement this system:

- **The Personal Computer**

A personal computer was used as the development and testing machine. The Personal computer (PC) running on windows 10 operating system, Intel(R) Celeron(R) N3350 Processor, a processor speed of 2.50GHz, Random Access Memory(RAM) size of 4.00GB, 64-bit Operating System and x64-based processor made the development easier and flexible, thereby making multi-tasking possible.

- **Aptana Studio**

Aptana Studio which is an open source integrated development environment (IDE) for building web applications based on Eclipse. This was used for coding the new system. It made coding in PHP easy and flexible. The reasons for the choice of Aptana Studio in the development of the new system are: Syntax Coloring according to the selected theme in the preferences, Code Assist, Syntax error annotations, Auto indentation and Code Formatting, Hyper-linking to classes, functions and variables by hovering over elements and pressing the Ctrl key, User friendly coding environment, PHPDoc popups when hovering over items that have attached documentation as well as Read and write occurrences markers when clicking on specific PHP elements.

- **Bootstrap 3**

During the development phase of the new system, BOOTSTRAP 3 was used to cascade the content aggregator platform. This was possible because it contains HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) based design templates for typography, buttons, navigation and other interface components, as well as optional JavaScript extensions. The reason for the choice of BOOTSTRAP 3in the development of the new system was because of the following features: Stylesheets, Re-usablecomponents, JavaScriptcomponents, JQuery,

- **XAMPP**

XAMPP was used to create a local web server for testing and deployment of the new system. It consists mainly of the MariaDBdatabase, interpreters for scripts written in the PHP, Perl-programming languages and Apache HTTP Server. XAMPP was chosen as a software tool for the testing and deployment of the new system because it is a simple, lightweight Apache distribution that makes it extremely easy to create a local web server for testing and deployment purposes.

- **PHP**

PHP recursive acronym for *Hypertext Preprocessor* is the scripting language used to program the new system; the reason for this is that PHP is especially suited for web development and can be embedded into HTML. PHP was preferred in the development of the new system. The PHP code is enclosed in special start and end processing instructions<?php and ?> that allows for easy navigation into and out of the "PHP mode".

- **PHP- Crawler**

This package provides a class to crawl links on a website. The PHP-Crawler was used as the crawling script to implement the web crawler so as to "spider" websites and pass information about all found documents (pages, links, files and so on). This framework for crawling and spidering websites was used to integrate the stochastic selector and incremental web crawling technology.

3.2 Method

The method involved the application of the neural network learning algorithm and engineering principles to integrate infospider and incremental web crawling technology in the development of a content aggregator. This is to enable the system to perform the desired processes with high level of efficiency. Acting out the Infospiders crawling method, the new system uses the back-propagation algorithm for learning. Such a learning technique provides Infospiderss with the unique capability to adapt the link, following behavior in the course of a crawl by associating relevance estimates with particular patterns of keyword frequencies around the links. The Infospiders Crawling Method is an improvement of the naïve best-first crawling method used by G. Pant et al (2004). The presence of the Infospiders crawling algorithm allows the news aggregator to search for pages through agents, relevant to the categories on the aggregator platform. Each agent is essentially following the crawling loop while using an adaptive query list and a neural net to decide which links to follow.The agent uses a stochastic selector to pick one of the links in the frontier. The algorithm provides an exclusive frontier for each agent. In a multi-threaded implementation of Infospiders, each agent corresponds to a thread of execution. Hence, each thread has a non-contentious access to its own frontier.

Also, the new system uses the incremental crawling algorithm to crawl the entire seed URLs. This is done continuously as downloaded pages are always updated, instead of starting from afresh. It is fast, has high freshness and low peak loads. As we know, continuous crawling requires the crawler to revisit the same resources at certain intervals which means that some intelligent control i.e. a record of each resource's history is used in turn to determine its ordering in a priority queue of resources waiting to be fetched. Here, an incremental strategy is called for rather than the snapshot strategy used in broad and focused crawls. This type of crawler is the one which on an incremental basis updates its collection of crawled contents once its accumulation of target is finally obtained. Subsequently, the existing collection is refreshed by performing new updates periodically.

4 TEST, RESULT AND DISCUSSIONS

The results obtained are presented in terms of the functions performed by the system. The news aggregator system was developed with a user-friendly interface. Therefore, it permits the interactions between human beings and the computer systems. The interface design is void of any form of ambiguity. The home page of the news aggregator is shown in figure7 below.

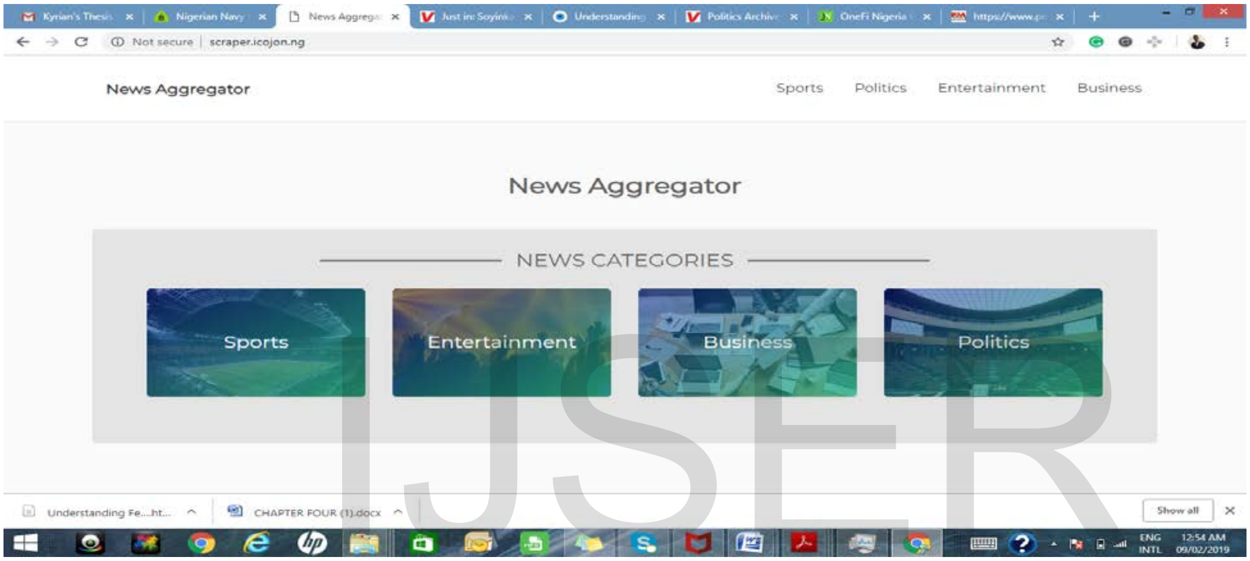
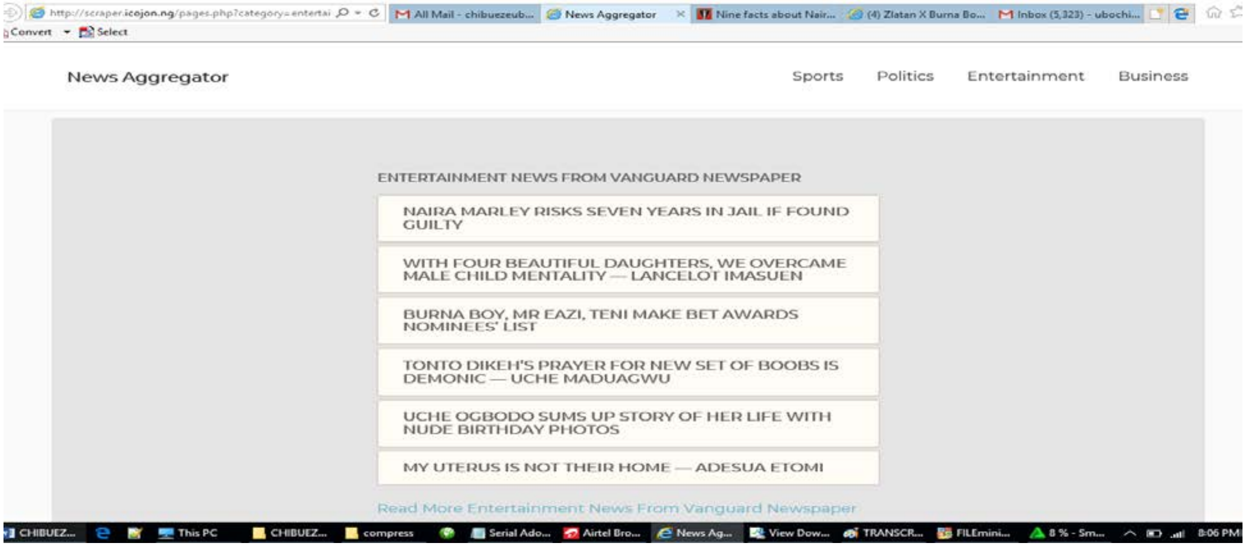


figure7: The aggregators' home page

The page shown in figure7 displays the section where the categories are presented to the users. Recall that these categories shown in the figure7 are the same categories fed as input into the inforspider representation to guide the URL frontier. The crawled output from authoritative news sites have been saliently organized into categories. The user is expected to select a category of interest either by clicking on the images or the menus at the top right corner of figure7. To always return to the home page, the user is expected to click on the "News Aggregator" menu at the top left corner of the same figure.

For instance, when the Politics category on figure7 is selected, the crawlers output shows the fetched latest news headlines under their respective authoritative news sites.

Figure8 below shows the latest aggregated Entertainment news from Vanguard newspaper.



ENTERTAINMENT NEWS FROM VANGUARD NEWSPAPER

NAIRA MARLEY RISKS SEVEN YEARS IN JAIL IF FOUND GUILTY

WITH FOUR BEAUTIFUL DAUGHTERS, WE OVERCAME MALE CHILD MENTALITY — LANCELOT IMASUEN

BURNA BOY, MR EAZI, TENI MAKE BET AWARDS NOMINEES' LIST

TONTO DIKEH'S PRAYER FOR NEW SET OF BOOBS IS DEMONIC — UCHE MADUAGWU

UCHE OGBODO SUMS UP STORY OF HER LIFE WITH NUDE BIRTHDAY PHOTOS

MY UTERUS IS NOT THEIR HOME — ADESUA ETOMI

Read More Entertainment News From Vanguard Newspaper

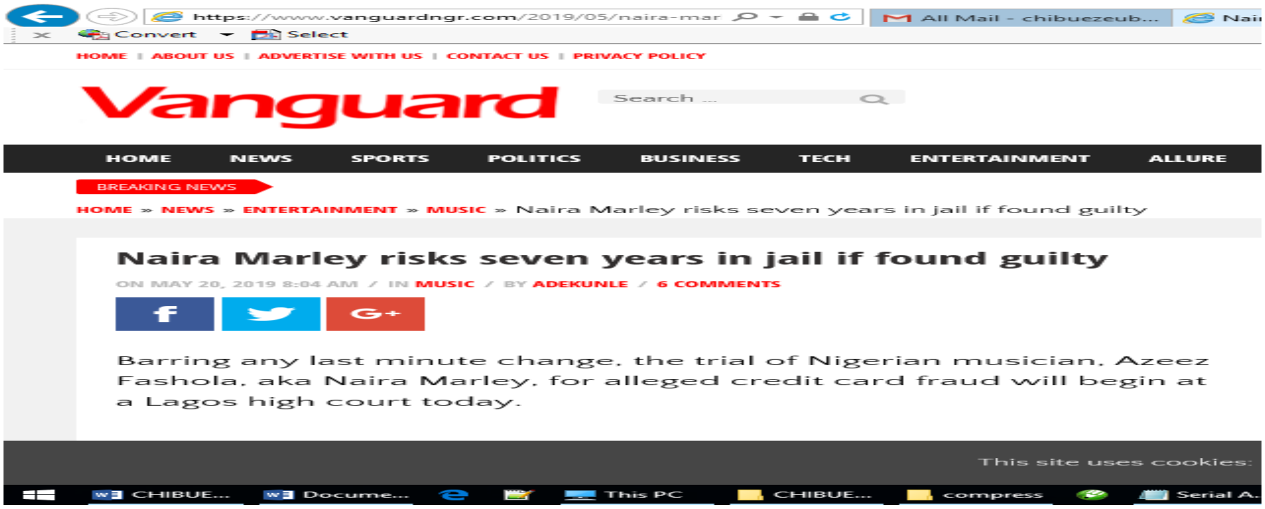
Figure8: Aggregated Entertainment news from Vanguard newspaper.

More aggregated Entertainment news from Vanguard newspaper can be viewed by clicking on the "Read More Entertainment News From Vanguard Newspaper". To access more click on the "Next Page".

Similarly, when either of the sport, entertainment or business categories are selected the crawlers output shows the fetched latest news headlines under their respective authoritative news sites. Taking a look at figure8, you will discover that the crawled outputs are well arranged, outputs of different authoritative (seed urls) sites are not jam-packed.

To view the outputs of other seed urls, the user needs to scroll down. To read any of the headlines of interest, the user selects or clicks on the aggregated headlines.

For example, in figure8 one of the headlines reads "NAIRA MARLEY RISKS SEVEN YEARS IN JAIL IF FOUND GUILTY". When the headline is selected the user is taken to the news site where he reads the news in detail as shown in Figure9 below:


https://www.vanguardngr.com/2019/05/naira-mar
All Mail - chibuezeub...
Nai
Convert
Select
HOME | ABOUT US | ADVERTISE WITH US | CONTACT US | PRIVACY POLICY
Vanguard
Search ...
HOME
NEWS
SPORTS
POLITICS
BUSINESS
TECH
ENTERTAINMENT
ALLURE
BREAKING NEWS
HOME » NEWS » ENTERTAINMENT » MUSIC » Naira Marley risks seven years in jail if found guilty
f
G+
This site uses cookies:
CHIBUE...
Docume...
This PC
CHIBUE...
compress
Serial A.

Naira Marley risks seven years in jail if found guilty

ON MAY 20, 2019 8:04 AM / IN MUSIC / BY ADEKUNLE / 6 COMMENTS

Barring any last minute change, the trial of Nigerian musician, Azeez Fashola, aka Naira Marley, for alleged credit card fraud will begin at a Lagos high court today.

Figure9: Detailed news of the selected news headline

5. CONCLUSION

An intelligent web based dynamic news aggregator integrating infospider and incremental web crawling technology has been developed using PHP scripting language to access PHP-crawler using Aptana Studio as IDE, Bootsrap3 and jQuery were used to provide a set of style sheets and JavaScript library to simplify the client-side scripting.

This work has shown the use of incremental and infospider web crawling technology to create an intelligent web based dynamic news aggregator, by consolidating many news websites into an aggregator. The system incrementally refreshes the existing collection of web pages based on the next threshold timestamp. It solves the problem of freshness of a page by employing the Infospider representation, which consisted of categories and a feedforward neural network to monitor initialized seed urls (authoritative Nigerian news sites) fed into the URL frontier for update via a breath-first traversal of the web. This provides a digital platform for individuals to easily find news pertaining to a particular news category in real time. Therefore, this system solves the problem of visiting numerous websites while memorizing different URLs to visit, all in a bid to view a specific type of news.

REFERENCES

[1] Chris Manning and Pandu Nayak (2008). Information Retrieval and Web Search. (Cambridge University Press, 2008). This book is available from Amazon, (FSNLP) Foundations of Statistical Natural Language Processing.

[2] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, (2008) Introduction to Information Retrieval, Cambridge University Press. 2008.

[3] Christopher Olston and Marc Najork (2010) "Web Crawling", now the essence of knowledge, Vol. 4, No. 3,2010. drudge-report-washington-times_n_861231.htm. lT) the Washington Post.

[4] F. Menczer, G. Pant, and P. Srinivasan (2003). Topical web crawlers: Evaluating adaptive algorithms. To appear in ACM Trans. on Internet Technologies, 2003. http://dollar.biz.uiowa.edu/~¯l/Papers/TOIT.pdf.

[5] F. Menczer, G. Pant, and P. Srinivasan. (2003). Topical web crawlers: Evaluating adaptive algorithms.To appear in ACM Trans. on Internet Technologies.

[6] G. Pant and F. Menczer. (2002) MySpiders: Evolve your own intelligent Web crawlers. Autonomous Agents and Multi-Agent Systems, 5(2):221{229, 2002.

[7] G. Pant, P. Srinivasan, and F. Menczer. (2002). Exploration versus exploitation in topic drivencrawlers. In-WWW02 Workshop on Web Dynamics, Honolulu, Hawaii.

[8] Gautam Pant, Padmini Srinivasan and FilippoMenczer, (2003) "Crawling the Web", The University of Iowa, Iowa City IA 52242, USA.

[9] Gautam Pant, Padmini Srinivasan, and Filippo Menczer (2004. Crawling the Web. School of Library and Information ScienceThe University of Iowa, Iowa City IA 52242

[10] Marc Najork and Allan Heydon (2001) High PerformanceWeb crawling. http://www.research.compaq.com/SRC/

[11] Monica Peshave (2017) HOW SEARCH ENGINES WORK AND A WEB CRAWLER APPLICATION. Article with 76 Reads. Cite this publication · https://www.researchgate.net/.../265994874_HOW_SEARCH_ENGINES_WORK_AND...

[12] *Otto, Mark (2012).* "Bootstrap in A List Apart No. 342". *Mark Otto's blog*. Archived *from the original on October 28, 2016*. Retrieved February 23, 2017.

[13] Rinki Tyagi (2016). Review of Ranking Algorithms. International Journal of Innovative Research in Computer and Communication Engineering(An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 4, April

2016 Copyright to IJIRCCE DOI: 10.15680/IJIRCCE.2016. 04041645205.

Ihekweaba Chukwugoziem *is a Professor in the Department of Computer Engineering, Michael Okpara University of Agriculture, Umuahia, Abia State, Nigeria. His research interests include Computer Hardware design and maintenance, Security system design, Digital Communication, etc.*
Email: Ihekweaba.gozie@mouau.edu.ng

Ubochi, Chibueze Nwamouh *is currently pursuing Masters in Computer Engineering, Michael Okpara University of Agriculture, Umudike, Umuahia, Abia State, Nigeria. His research interest include Software and Systems Engineering, Information Management System and System Programing.*
Email: chibuezeubochi@gmail.com

Aru, Okereke Eze *is a Senior lecturer in the Department of Computer Engineering, Michael Okpara University of Agriculture, Umuahia, Abia State, Nigeria. His research Interests include Computer Hardware design and maintenance, digital systems design using microcontrollers and other computer related subjects.*
Email: okezearu@yahoo.com