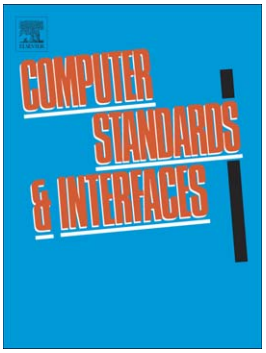

Empirical Evaluation of the Link and Content-Based Focused Treasure-Crawler

Ali Seyfi, Ahmed Patel

PII: S0920-5489(15)00104-X
DOI: doi: 10.1016/j.csi.2015.09.007
Reference: CSI 3065

To appear in: *Computer Standards & Interfaces*

Received date: 21 April 2015
Revised date: 19 July 2015
Accepted date: 14 September 2015

Please cite this article as: Ali Seyfi, Ahmed Patel, Empirical Evaluation of the Link and Content-Based Focused Treasure-Crawler, *Computer Standards & Interfaces* (2015), doi: 10.1016/j.csi.2015.09.007

Empirical Evaluation of the Link and Content-Based Focused Treasure-Crawler

Ali Seyfi [a,*] and Ahmed Patel [b,c]

[a] Department of Computer Science, The George Washington University, Washington DC, United States
[b] Faculty of Computer Science and Information Systems, Jazan University, Saudi Arabia
[c] Visiting Professor, Faculty of Science, Engineering and Computing, Kingston University, United Kingdom

Abstract

Indexing the Web is becoming a laborious task for search engines as the Web exponentially grows in size and distribution. Presently, the most effective known approach to overcome this problem is the use of focused crawlers. A focused crawler employs a significant and unique algorithm in order to detect the pages on the Web that relate to its topic of interest. For this purpose we proposed a custom method that uses specific HTML elements of a page to predict the topical focus of all the pages that have an unvisited link within the current page. These recognized on-topic pages have to be sorted later based on their relevance to the main topic of the crawler for further actual downloads. In the Treasure-Crawler, we use a hierarchical structure called T-Graph which is an exemplary guide to assign appropriate priority score to each unvisited link. These URLs will later be downloaded based on this priority. This paper embodies the implementation, test results and performance evaluation of the Treasure-Crawler system. The Treasure-Crawler is evaluated in terms of specific information retrieval criteria such as recall and precision, both with values close to 50%. Gaining such outcome asserts the significance of the proposed approach.

Keywords: Focused Crawler, Topical, T-Graph, HTML Data, Information Retrieval, Search Engine

1. Introduction

To improve the quality of searching and indexing the Web, our proposed focused crawler depends on two main objectives, namely, to predict the topic of an unvisited page, and to prioritize the unvisited URLs within the current page by using a data structure called T-Graph. We elaborated the architecture of the Treasure-Crawler in a preceding paper [1], where a thorough review on the background of this subject field was discussed by briefly introducing significant Web crawlers. Also, the requirements of a focused crawler were elicited and the evaluation criteria were outlined.

Based on the current enormous size of the Web, which has passed sixty trillion Web pages [1], the need for the distribution of resources to harvest, download, store and index the Web has become a clear necessity. In this context, focused crawlers are inevitable tools, aimed for detecting pages based on the diverse topics of human knowledge, and indexing the relevant pages into distributed repositories while keeping a merged index for further faster retrievals.

This task of indexing must be capable enough to keep up with the changes of dynamic contents on the Web, since keeping an old version of the Web is obsolete. That is the reason for employment of machine learning or reinforcement learning strategies within the body of some search engines; to diligently watch the Web. Section 2 includes a succinct background study of focused crawlers. Section 3 is the introduction to our proposed prediction and prioritization methods. Then the implementation and experimental setup details and results are discussed in Section 4. This paper concludes with the evaluation of the results and the future directions for interested researchers.

---

[*] *Corresponding author*

E-mail addresses: seyfi@gwu.edu (A. Seyfi), whinchat2010@gmail.com (A. Patel)

2. Background and Requirements

To cover the maximum possible number of Webpages, the first generation of crawlers recursively follow the links between Web documents. Therefore, these exhaustive crawlers pay less attention to page content as opposed to the interlinked structure of the Web. The exponential growth of the Internet motivated the scientists to design focused crawlers that restrict their concentration to specific topics and as a necessity value the page content in addition to the link structure between pages. In this section, we introduce the most appropriate of these focused crawlers as state of the art. Then in Section 6 we present a high level comparison between their approaches and our proposed system in terms of used algorithms, modules, and concepts. In terms of performance, one of these crawlers is compared to the Treasure-Crawler.

Context Focused Crawler (2000) has been the basis for many crawlers during the past decade. An exemplary multi-level tree is built out of sample documents for each target Web page. These graphs are then merged into one and help prioritizing any downloaded page by finding its content's similarity to the nodes in the graph [2].

Meta Search Crawler (2004) relies on the results from other search engines, where several topic-specific queries are sent to multiple search engines and the results are merged and indexed. This approach has a good accuracy in remaining on topic, while having the precision value of 50.23%, although its dependence on other search engines is questionable [3].

LSCrawler (2006) uses the corresponding ontology of its topical focus, and semantically assesses the anchor text and the text around any unvisited URL. This crawler has its recall value as 0.6 reportedly, which is very close to that of the Treasure-Crawler, though it is based on semantic technology [4].

Relevancy Context Focused Crawler (2006) addresses the high classification load and low accuracy issues of the original context graph algorithm [2] and uses a novel structure that reaches 22% in precision. This approach uses the general and on-topic feature words to enhance the prioritizing process [5].

Hybrid Focused Crawler (2006) is based on both content similarity and link structure (links-to and links-from). The content similarity is calculated using vocabulary vectors and simple keyword matching. This crawler demonstrates a high harvest ratio in the retrieval of on-topic pages since it is intended to first download and then determine the relevancy of a Web page [6].

HAWK (2008) is another known link and content based combinatory method to focused crawling the Web. HAWK has the harvest ratio of 60% significantly decreases after traversing 5000 pages as reported [7].

Modified Naïve Bayes (2010) is the root classification technique for a crawler that uses reinforcement learning to be trained on the topic taxonomy. It also uses a modified form of TF-IDF for its assessments of textual data. These modifications bring about a relatively high harvest ratio and performance [8].

Intelligent Focused Crawler (2011) carries out the URL scoring task based on a threshold-restricted modified form of Naïve Bayes classifier. The modifications resulted in a significantly improved harvest ratio of 80%, although the number of crawled pages in the experiment (1000 pages) is not a reliable size for test dataset [9].

OntoCrawler (2011) employs a fuzzy technique to weigh the content of a page and is considered as a semantic crawler, as it relies on ontologies for knowledge classification. OntoCrawler has empirically shown the precision rate of 90% on the topic of football, which is a significant performance [10].

Clickstream-Based Parallel Crawler (2012) follows the history clickstream parameter as opposed to interlink structure of Web pages for the URL prioritization task. Therefore, an authorized access to server log is required as this approach relies on previous users' activity. We consider this technique as a potential module to be added to our proposed system [11].

BDS for SIS Crawlers (2014) is a novel technique called Bridge Driven Search and addresses the Social Internetworking Systems. In this flexible technique, the problem of loose connections between thick populations of on-topic regions on the Web is targeted. In our proposed system, we solved this problem with appending off-topic pages to the download queue with the lowest priority score, while making certain that their priority score is incremented gradually [12].

Beam Classifier (2014) is based on decision trees for the classification of lexicons. However, this lexicographic classifier shows more accuracy as compared to the greedy approach of DTs. Also a fitness function constantly watches the accuracy and F-measure to evaluate the performance. We are now studying the possibilities of incorporating a similar solution into the future version of our Treasure-Crawler, with respect to the fact that using trie data structure will result in a better performance because of its linear time complexity [13].

While most of the research in this area is carried out in the scope of designing better classification and link scoring functions, some scientists set out their software engineering viewpoint as the basis of their research,

taking into account the programming, implemental, and applicative details of such crawling systems [14, 15]. Some of these considerations are specific features of Web pages such as DOM, Internet applications' automatic crawling, guided and example-based approaches to crawling, cyber-security threats and system vulnerabilities, and Web application development methods such as component-based technique. It is noteworthy that based on software engineering standards, we studied the essential requirements of a Web crawler system [1]. These requirements are grouped into functional requirements, such as relevance calculation, priority calculation, and data storage and indexing for later search. The non-functional requirements are more abstract and include flexibility, robustness, standardization, manageability, modularity, etc.

3. Prediction and Prioritization Methods

In our preceding paper [1], we presented the architecture of a Treasure-Crawler-based search engine and its modules. Predicting the topical focus of an unvisited page and assigning appropriate priority score to its URL are the main functions of a focused crawler.

For the topic prediction task, we designed a custom method to detect the main subject(s) of the target page. In order to best classify the main subject of a Web page into known and standard human knowledge categories, we use the Dewey Decimal Classification (DDC) system [16, 17] as the human knowledge classification reference. Figure 1 shows an example of finding the topic 'butterfly' in DDC:

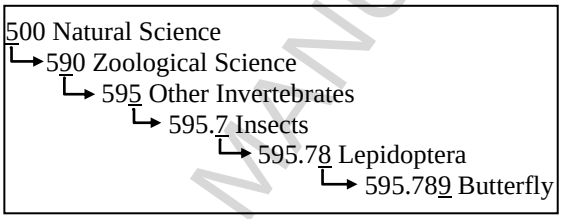

Figure 1. Finding the topic 'butterfly'

A dataset is created from the vocabulary and corresponding codes from DDC (see Table 1 in the next Section). Basically, the word tokens from specific HTML elements of the parent document are extracted, preprocessed, cleaned, and then compared to the entries of this dataset in consecutive phases, each phase corresponding to a digit position of the code, i.e. 1st digit, 2nd digit, and so on. In each phase of the prediction process, the acquired topical focus becomes more precise and detailed. If the final selected codes (indicating the predicted topic) are parts of the desired DDC list of codes, then the unvisited page is considered as on-topic [1].

For the prioritization objective and in addition to the above procedure, the T-Graph structure as an exemplary guide carries out the task of priority association by providing a conceptual route for the crawler to follow and find on-topic regions. T-Graph is a tree-like hierarchical structure where each node has possibly multiple parents in multiple upper levels, and evidently each node has multiple children in any lower levels, while the target (most relevant) documents are in the lowest level (leaves). The tokens of HTML elements of the parent page along with the anchor text of the unvisited link are compared to all nodes of the T-Graph, the node with highest similarity is found and its level number in the tree indicates the priority score of the unvisited link, calculated as: (1/level_number). The URL of the unvisited page is then inserted into the download queue (i.e. a priority queue) along with this priority score for later download.

Based on these objectives, we designed and implemented our system as a *prototype*. From the evaluated results of our experiment, we can assert that the framework, architecture, design, and implementation of our prototype were effective and satisfied the two objectives of this research. The rest of this paper elaborates the experimental setup and results.

4. Experimental Setup

The Treasure-Crawler is designed and implemented based on the modularity and object oriented concepts and techniques; hence all the modules can simply be plugged and played while requiring the minimum change of other modules and at interface level only. The prototype experiment is conducted to test the capability of the crawler to download Web pages related to a predefined topic/ domain, by best using the T-Graph structure and other methods that are defined or used in this research. With this overview, the Treasure-Crawler system was setup as elaborated in Figure 2, showing the main components as well as the data that travel within the system, with the following explanation:

The *crawler robot* contains fetcher and response queues. The seed URLs are initially inserted into the *fetcher queue* with the highest priority. When an item (URL) with the high priority is fetched (1), the *robot* downloads the textual content of the page from the Internet (2). The HTML data of the page, or if unsuccessful, server's HTTP response relating to temporary unavailability of the link, is then transferred to the *structural HTML parser* (3) where specific HTML elements are extracted and given to the *crawler* (4). These elements are properly segregated and then given to the *relevance calculator* (5) to algorithmically predict the topical focus of all unvisited links within the page. For this task, the *relevance calculator* is supplied with the complete set of *DDC entries* (6), as well as the set of DDC entries that specify the crawler's *topic of specialty* (7). If the *relevance calculator* determines an unvisited link as on-topic, the data from specific HTML elements are compared to *T-Graph* nodes (8) and the priority score of the link is calculated (9). If the unvisited link is determined as off-topic, it receives a lowest priority score in order to push the crawler into harvesting unconnected on-topic regions of the Web. The URL along with its priority score is then inserted into the *fetcher queue* (10), where the score of items is cyclically incremented to prevent starvation. Also, the actual HTML data of the current page is stored in the *repository* (11) with other measurements such as the priority scores given to the links. The rest of this section is on the description of how the test is carried out.

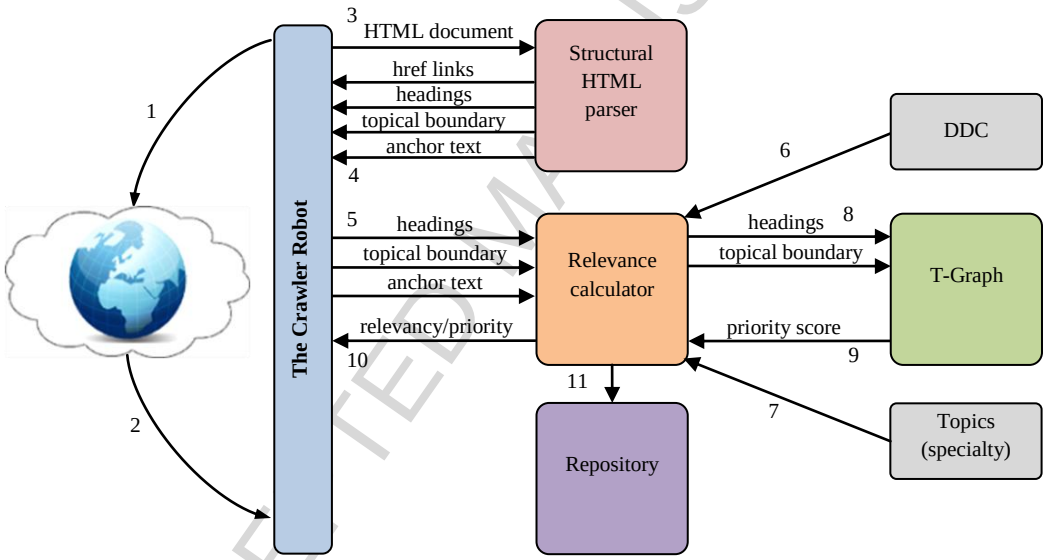

Figure 2. Crawler robot setup for the experiment

4.1 Topics

The system has been manually supplied with DDC codes and corresponding word tokens relevant to the topic of English language and grammar, in which the system specializes. It must be mentioned that DDC is a proprietary dataset of codes and classes, organizing human knowledge into distinct categories. All the textual data is lowercased and stemmed using the Porter stemming algorithm before taking part in any calculation or comparison. This way, the acronyms of each word will be taken as similar to the base word. Table 1 lists the DDC data entries indicating the topical focus of the system, i.e. English language and grammar.

Table 1. Base DDC codes and classes

No.	Code	Class (Initial)
1	400	Language
2	403	Dictionaries & encyclopedias
3	410	Linguistics
4	412	Etymology
5	413	Dictionaries
6	414	Phonology & phonetics
7	415	Grammar
8	417	Dialectology & historical linguistics
9	418	Standard usage & applied linguistics
10	419	Sign languages
11	420	English & Old English
12	421	English writing system & phonology

13	422	English etymology
14	423	English dictionaries
15	425	English grammar
16	427	English language variations
17	428	Standard English usage
18	429	Old English (Anglo-Saxon)
19	490	Other Languages
20	820	English & old English literatures

4.2 Seeds

Seeds are the designated starting points of the system for crawling. These points, in terms of URLs, are initially handed to the crawler to start populating its document collection. These seed URLs as well as the topics of interest constitute the essential parameters of a Web crawler. In our experiments, two different sets of seeds were supplied to the system. The first set consisted of seventeen *on-topic* URLs, while the second set contained seven *generic* URLs, as listed in Table 2.

Table 2. On-topic and generic seed URLs

NO.	URL	Generic(G)/ On-Topic(T)
1	http://www.englishclub.com/...	T
2	http://jc-schools.net/...	T
3	http://www.brighthub.com/...	T
4	http://www.englishforum.com/...	T
5	http://tls.vu.edu.au/.....	T
6	http://onlineenglishhub.blogspot.com/	T
7	http://openlearn.open.ac.uk/.....	T
8	http://englishhubonline.net/...	T
9	http://muse.jhu.edu/...	T
10	http://www.mla.org/...	T
11	http://robin.hubpages.com/...	T
12	http://www.scientificpsychic.com/...	T
13	http://home.pacific.net.au/...	T
14	http://english.berkeley.edu/...	T
15	http://englishplus.com/...	T
16	http://esl.about.com/.....	T
17	http://www.onlineenglishdegree.com/...	T
1	http://www.dmoz.org/	G
2	http://dir.yahoo.com/	G
3	http://vlib.org/	G
4	http://www.stpt.com/...	G
5	http://www.joeant.com/	G
6	http://botw.org/	G
7	http://www.ansearch.com/...	G

4.3 T-Graph

As described before, the T-Graph has the possibility of being constructed either top-down or bottom-up. According to the requirements of our implementation, the construction of T-Graph starts from a set of target documents that are the most topically relevant Web pages, and from that point onwards, their parents (in terms of Web hyperlinks) are retrieved. Therefore, the utilized approach for T-Graph construction is bottom-up. The target documents in our experiment are:

- http://en.wikipedia.org/wiki/Language
- http://en.wikipedia.org/wiki/Grammar
- http://www.reference.com/browse/language
- http://www.reference.com/browse/grammar

Using a backlink checker tool, the Open Site Explorer [18], the parents of these four target documents are retrieved to form the nodes of level 1. In the same fashion, the nodes of levels 2 and 3 are retrieved and for each

node the constituent components are filled with the necessary data. In total, the graph has fifty nodes, made out of twenty eight URLs. These nodes constitute the graph in four levels (0-3). It must be stated that several edges are discarded in this implementation of the graph. Table 3 lists the URLs that are used in the T-Graph, as well as number of nodes made out of each.

Table 3. URLs used in our actual T-Graph and their corresponding number of nodes

No.	URL	Node Count
1	http://en.wikipedia.org/wiki/Language	1
2	http://en.wikipedia.org/wiki/Grammar	1
3	http://www.reference.com/browse/language	1
4	http://www.reference.com/browse/grammar	1
5	http://en.wikipedia.org/wiki/Linguistics	2
6	http://en.wikipedia.org/wiki/Translation	1
7	http://en.wikipedia.org/wiki/Natural_language	2
8	http://en.wikipedia.org/wiki/Sentence_(linguistics)	2
9	http://en.wikipedia.org/wiki/Word	2
10	http://en.wikipedia.org/wiki/Poetry	2
11	http://www.reference.com/browse/linguistics	1
12	http://www.reference.com/browse/speech	1
13	http://en.wikipedia.org/wiki/Syntax	2
14	http://en.wikipedia.org/wiki/Drama	2
15	http://en.wikipedia.org/wiki/Literature	2
16	http://en.wikipedia.org/wiki/Writing	2
17	http://en.wikipedia.org/wiki/Etymology	2
18	http://en.wikipedia.org/wiki/Philosophy_of_language	2
19	http://www.reference.com/browse/anthropology	1
20	http://www.reference.com/browse/mouth	1
21	http://en.wikipedia.org/wiki/Communication	4
22	http://en.wikipedia.org/wiki/Sign_language	4
23	http://en.wikipedia.org/wiki/The_arts	2
24	http://en.wikipedia.org/wiki/Programming_language	3
25	http://en.wikipedia.org/wiki/Anthropology	2
26	http://en.wikipedia.org/wiki/Genre	2
27	http://www.reference.com/browse/ethnology	1
28	http://www.reference.com/browse/dentistry	1

4.4 Experimental Environment

The system is run with different sets of seeds, configurations and conditions in order to be well evaluated. All the runs are carried out on commonly configured machines with permanent Internet ADSL connection. As described before, our experimental prototype system is specialized in the topic of English language and grammar. The April 2014 version of DMOZ directory [19] is employed as the dataset, with the metadata shown in Table 4. It must be stated that the DMOZ dataset is originally an ontology. For our system, the converted version of the DMOZ dumps has been used, which is in MySQL database format.

Table 4. Metadata of DMOZ dataset

Description	Record Count
Total number of URLs	3,922,814
Total number of topics	771,986
Number of on-topic URLs	10,404
Number of related topics	1,345

4.5 Tuned Running Conditions

We tested the system on a development set of data (a portion of the dataset that we use for trial runs to find best initial values for system parameters) to come up with a set of tuned parameters that produce highest functionality/performance. Table 5 lists these parameters along with their tuned values.

The 1st parameter is the depth of T-Graph that ultimately affects the priority score assigned to URLs. If the textual similarity of the selected HTML elements to some node(s) of T-Graph is above a threshold (2nd parameter) then the node at the lowest level is selected and its level number helps calculate the priority score of the URL as (1/level_number). It is noteworthy that a deeper T-Graph results in more accuracy in assignment of priority score. However in the scope of our experimental prototype, the depth of 3 will reasonably help distinguish highly topically-related pages. The 3rd item indicates that anchor text of a link has 40% more impact on the topic prediction of the unvisited page, as opposed to the text around the link. The 4th item specifies the number of digits of DDC codes (D-numbers) that the system processes to determine the topical focus. Obviously, as the number of processed digits becomes greater, the accuracy of the topic detection increases. The 5th item is the default assigned priority value to an unvisited link when it has no corresponding node in the T-Graph with OSM value of more than the threshold or when the topical focus of the unvisited link is not what the system specializes in. The 6th item is the value that is periodically added to the priority score of each item in the fetcher queue until it is less than 1 to prevent aging of the queue items (also called starvation). This value is added, after a specific number of insertions which is the 7th item in the table.

Table 5. Initial parameters default values

No.	Parameter	Value
1	T-Graph Depth	3.00
2	OSM Threshold	0.05
3	Anchor Text Impact Factor	1.40
4	Maximum D-number Length	3.00
5	Unrelated Priority	0.01
6	Fetcher Queue Aging Factor	0.05
7	Item Count to Apply Aging Factor	100.00

To evaluate the performance of the Treasure-Crawler, in addition to the default tuned parameters, we alternatively tested the system with changes in three individual parameters to observe and compare their impact on the performance. Table 6 lists these alternative conditions under which the system has been run and tested. For each set of parameters' values, we run the system with both Generic and On-topic seed URLs, listed in Table 2.

Table 6. Running conditions of Treasure-Crawler system

On-Topic Seeds	Generic Seeds	Parameter	Value
T1	G1	*All defaults*	N/A
T2	G2	OSM Threshold	0.10
T3	G3	Anchor Text Impact	0.50
T4	G4	Fetcher Aging Factor	0.02

5. Experimental Results

In our experiment, the fundamental criterion to be measured for the crawler was the number of retrieved on-topic pages during the runs. Concurrently, we tested the system against different input values (see Table 6) in order to improve our future versions of the Treasure-Crawler. It must be stated that not all the input parameters and their corresponding alternate values are given here in order to prevent the overemphasis on the system parameters. One other fact is that the performance results of the Treasure-Crawler are presented only based on the default values of the input parameters, while we observed that for some parameters, the alternate value yields a better performance. These values will be considered in further implementations of the Treasure-Crawler and presented in follow up papers.

Figure 3 shows the number of retrieved on-topic pages for runs with generic seed URLs and in each block of 1000 crawled pages. Figure 4 shows the same result for runs with on-topic seed URLs. Based on these results, it is observed that the number of retrieved pages for runs with on-topic seeds stays more satisfactory. This is because the crawler initially learns to discard the regions on the Web that yield the least number of relevant pages, and tries to go through the highly populated regions. Figures 5 and 6 show the cumulative

number of retrieved on-topic pages. The observable increase in this figure conveys the good performance of the Treasure-Crawler. Different effects of seed types are also apparent in these figures.

As seen in the following figures, T4 and G4 show a rather better performance. This shows that increasing the priority score of the fetcher queue items with a lower value will help related URLs to be fetched and downloaded faster, and hence their associated Web pages are primarily analyzed.

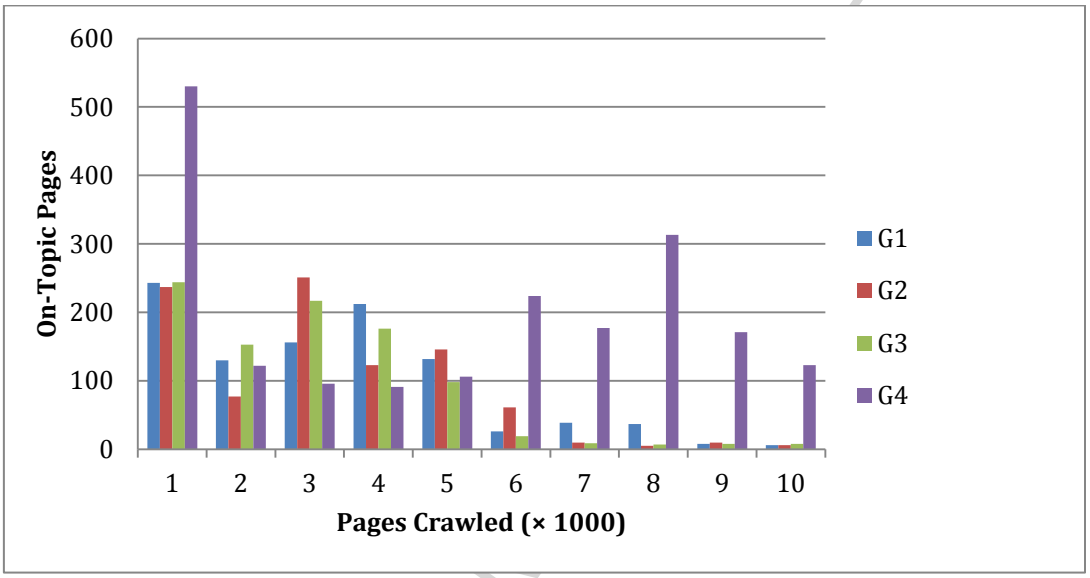

Figure 3. On-topic retrieved pages (Generic seeds)

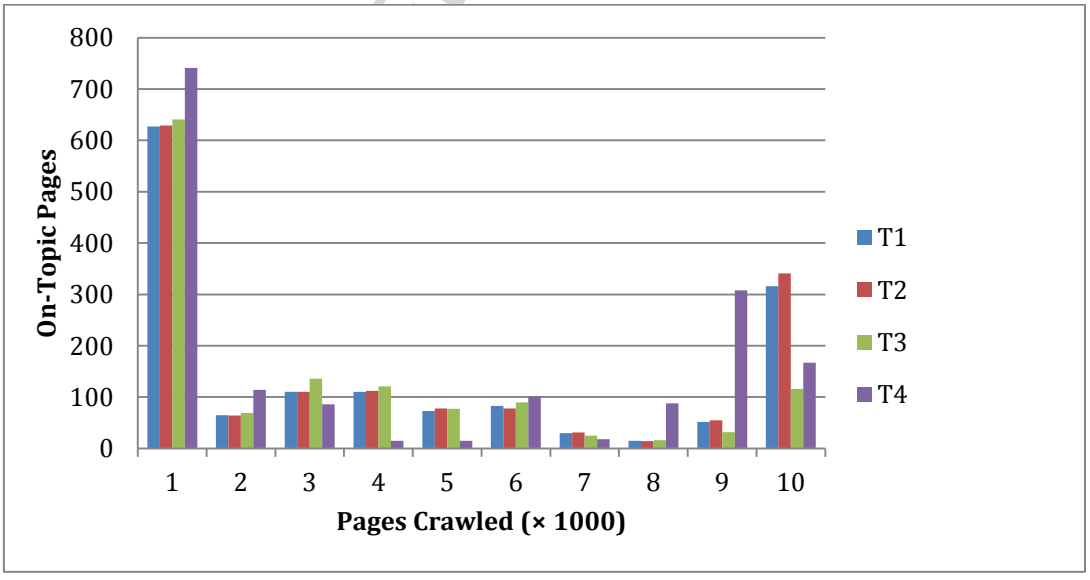

Figure 4. On-topic retrieved pages (On-topic seeds)

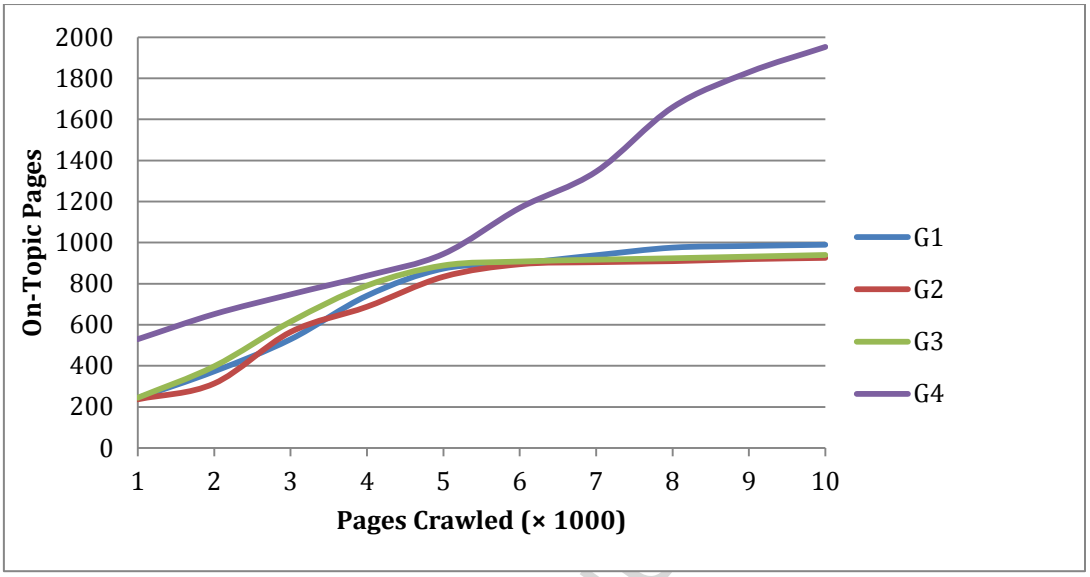

Figure 5. On-topic retrieved pages (Generic seeds)

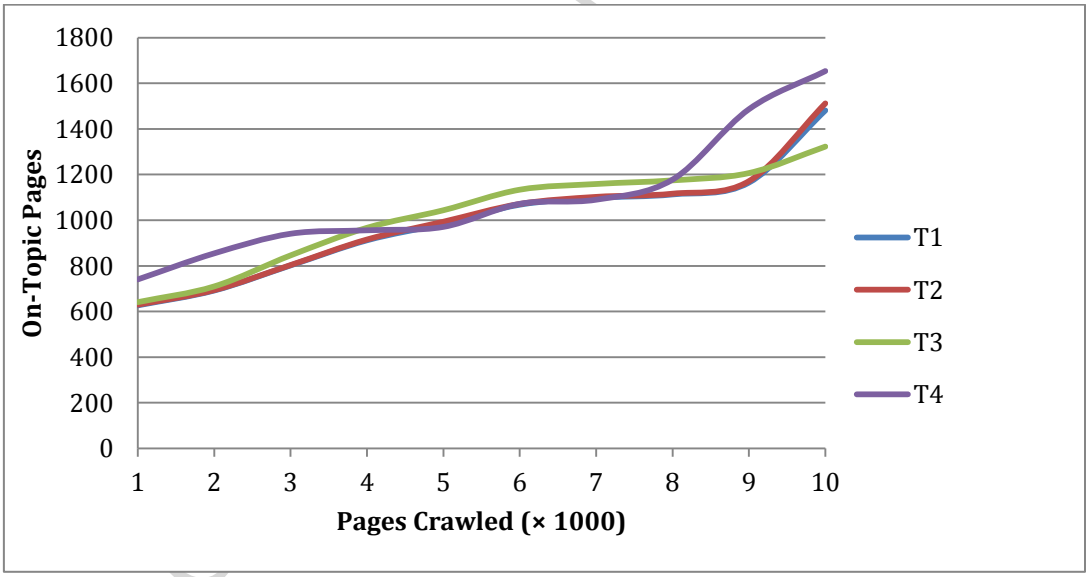

Figure 6. On-topic retrieved pages (On-topic seeds)

Harvest ratio is a significant IR performance metric, and is broadly used in the evaluation of focused Web crawlers. Although the Treasure-Crawler shows a decreasing harvest ratio, its gradual value is still reliable. Considering the effect of the fetcher queue aging factor (as in T4 and G4), the alternate values of other initial parameters, and other system and resource limitations in our research, the harvest ratio of the Treasure-Crawler remains acceptable. (see Figures 7 and 8)

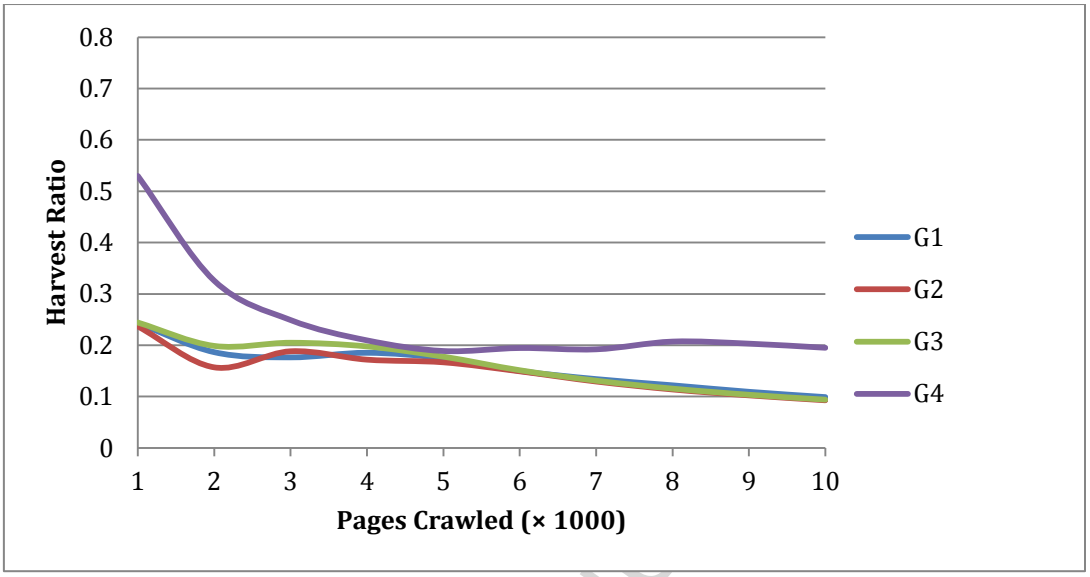

Figure 7. Harvest Ratio (Generic seeds)

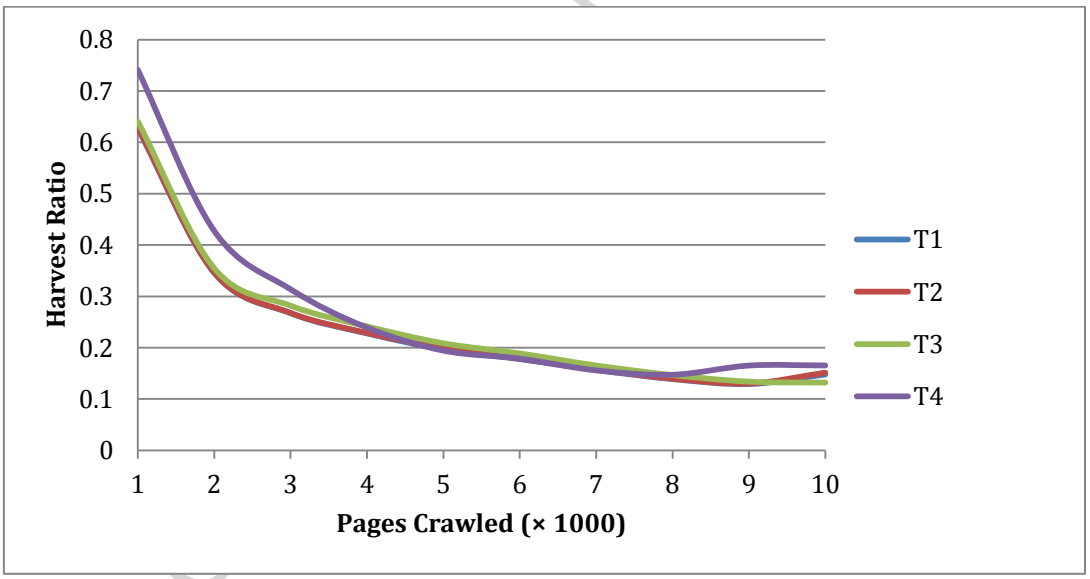

Figure 8. Harvest Ratio (On-topic seeds)

6. Evaluation of the Results

Generally a crawler is expected to retrieve "good" pages. However, to determine what is a *good* page is not an easy task. The performance of a focused Web crawler can be assessed in several aspects listed below for the sake of completeness. We carried out our evaluation based on precision, recall, and number of retrieved good (target) pages.

Page Importance: Basically two major approaches exist in defining an important page; link-based and similarity-based [20]. What we discuss in our focused crawler is similarity-based importance of a page. In this context, similarity is defined as the relevance score of the page to a specific topic in which the crawler specializes. However, measuring, tuning and setting thresholds for this score are experimental tasks. Also, the calculation algorithm of this metric is a fundamental factor in designing a good crawler and our proposed system shows a reasonable functionality in this regard.

Access Speed: The rate at which a crawler covers a part of the Web defines its speed. This speed depends heavily on the algorithms and data structures it uses. Generally, the average processing time to produce outcomes, whether positive or negative is the important factor to evaluate the speed. The methods used in our system have linear time complexity and the speed is not a concern.

Repository Freshness: The Web is rapidly growing and changing. Every Web page's content changes in an unpredictable time interval. To keep up with this change, the crawler has to provide its repository with an

algorithm or facility that brings freshness to its contents, meaning that periodically checks the downloaded pages for possible further changes.

Scoring Function: To provide an appropriate priority score for a Web page, the system should employ fast and optimal functions. Using the T-Graph in this regard is a jump towards the optimality of this score. Authors of the context graph algorithm, Diligenti et al. [2] have also shown that the average relevance of the downloaded on-topic documents remains consistent and improved, if a sliding window is employed to limit the number of downloads. This concept has not been tested in our system but is proved to improve the performance of such crawler systems.

Efficiency: A crawler requires several resources such as network bandwidth to crawl the Web and acquire documents, memory to keep its internal data structures to be used by its algorithms, processor to evaluate and select URLs, and disk space to keep the processed text and extracted links. In our prototype implementation, we followed specific software engineering guidelines to satisfy not only the focused crawler specific requirements but also those software system requirements to make an efficient system.

Recall and Precision: Recall and precision are the two significant performance measures. There are many other suggested metrics that depend on recall and precision in nature. For example, "the percentage of relevant pages retrieved over time" and "the percentage of target pages found while the percent of hyperlinks followed increases" are both estimations of recall. Also "the harvest rate" [21], "the average rank of retrieved pages" [20], and "the average relevance of pages" [2] are all estimates of precision. Srinivasan et al. [22] proposed a novel performance-per-cost resolution to measure the effectiveness of a crawler against its efficiency. The literature on the performance of focused Web crawlers demonstrates that in all approaches of crawling the Web, as the number of crawled pages grows, the recall and precision values act conversely. They also studied several crawling algorithms and have experimentally proven that as the number of crawled pages grows, the average recall of the target pages levels up, while the precision decreases. It must be noted that in a large number of pages, both recall and precision approach a constant level.

To present the performance of the Treasure-Crawler, it is compared to the context focused crawler [2], which has been the basis for many researches in the domain of Web search crawlers. This approach focuses on the link distance of each Web page from a target document, comprising that if a Web page corresponds (in terms of content similarity of the whole page) to a node in a layer of the graph, the priority of the unvisited link in that page accords to that layer of the graph.

Although the context graph relies on the assumption that the documents follow a common hierarchy, it is obvious that the Web pages are not well-organized and homogenously structured. This makes the use of a layered structure inevitable. Figure 9 compares the performance of the Treasure-Crawler and the context graph algorithm in terms of remaining on topic while crawling the Web. Our proposed system significantly outperforms the context graph algorithm in terms of the number of retrieved on-topic documents.

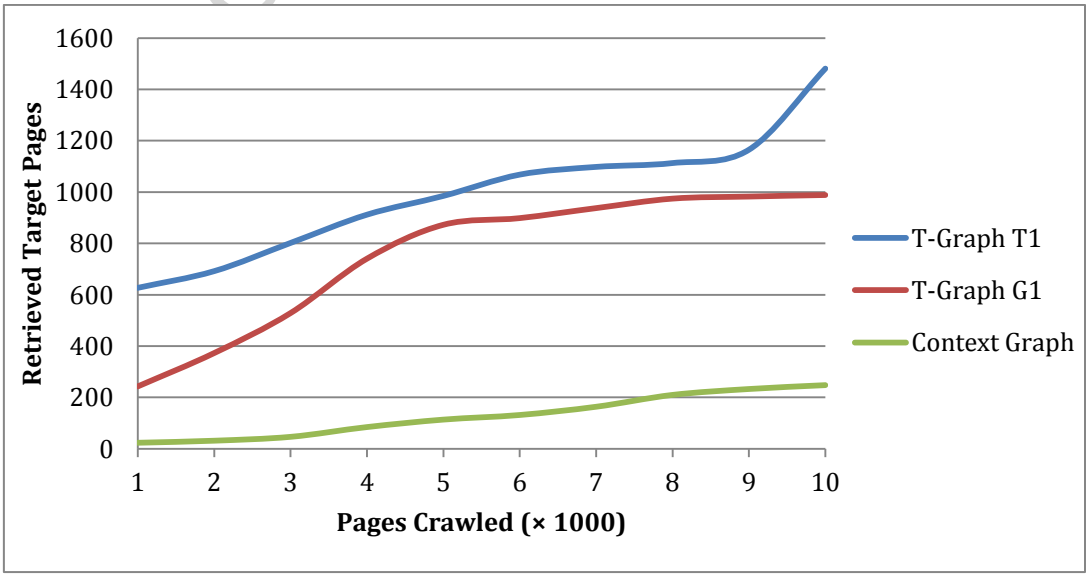

Figure 9. Comparison of retrieved documents

In order to obtain the ratio of the *relevant* to *all* crawled pages, the recall value (0.50) which is gained from running the system while linked to the DMOZ dataset is multiplied to the number of retrieved pages. Figure 10 shows the results, where our system still outperforms the context graph algorithm.

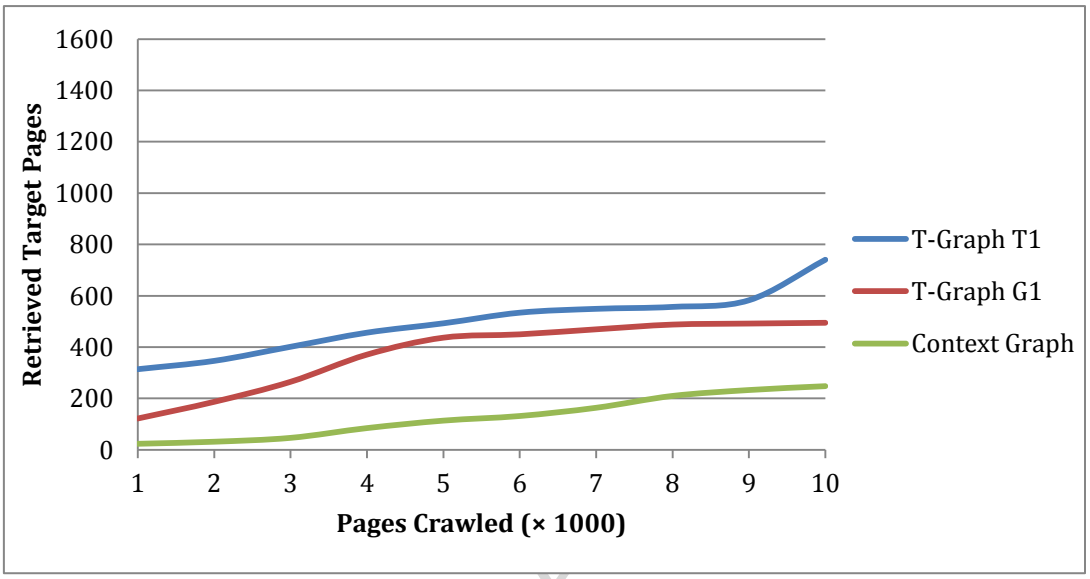

Figure 10. Comparison of retrieved documents multiplied by recall value (0.5)

In the vast domain of Web crawler technology, there are several common components and algorithms that are frequently used by the designers. Table 7 shows the relevance calculation and classification methods as well as experimental results of the surveyed algorithms. It must be stated that we have not re-used any implementation of these approaches for our evaluation task, and the numerical values in Table 7 are what the authors have originally reported. Although our proposed algorithm hardly uses these methods, they obviously have had a great influence on the crawlers so far. It can be vividly derived from Table 7 that the link-based algorithms are more widely used since downloading the entire content of a Web page puts a huge burden on the search engine server, as in content-based methods. The Treasure-Crawler employs a novel interpretation of the surrounding text by taking into account the major HTML elements of the page. The classification method, however, highly depends on how a possible approach is adopted. What makes the Treasure-Crawler innovative in this regard is that it totally relies on a custom model in classification and prioritization of pages. This characteristic affects the results to be lower than other crawlers but permits flexibility that could be enhanced in future versions. Also as observed from Table 7, harvest ratio and precision/recall act conversely. One significant reason for our harvest ratio to be below average is the limitations of DMOZ in terms of number of classified URLs.

Table 7. Comparison of different crawlers' relevance calculation and classification methods

Focused Crawler (FC)	Relevance Calculation					Classification				Results		
	Whole Content	URL	Anchor Text	Terms Around Links	Custom Method	Naïve Bayes	Decision Tree	Custom Model	Semantic	Recall	Precision	Harvest Ratio
Context FC	*					*	*					2.5%
LSCrawler		*	*	*				*	*	60%		
Relevancy CFC		*	*	*							22%	
Hybrid FC	*							*				67.2% - 85.5%
Meta Search					*			*			50.23%	
HAWK	*	*	*	*				*				60%
Intelligent FC					*	*						80%
Modified NB		*	*			*		*				33%
OntoCrawler	*						*	*	*		90%	
Treasure-Crawler		*	*	*	*		*	*		50%	50%	20%

Since all the named systems seem to have close results, it is definitely the matter of choosing the right method. The crawlers are composed of several components, for each of which the designer has many choices. Therefore, in order to make a better comparison among this number of systems so far, it is desirable that each fundamental component be compared/evaluated separately. Also, we lack a general evaluation framework with the capability of hosting any Web crawler, which can automatically test and evaluate the different parts of the crawler, namely the relevance calculator and the prioritizer.

In this section, we compared the results from our prototype system with the performance of the context focused crawler. Our system holds the value of 0.5 for both recall and precision, which implies that our proposed system outperforms the context focused crawler in terms of these two substantial criteria. Although many improvements have been made on the classic context graph algorithm, no one has tried to remove the strict link distance requirement to make this algorithm more adaptable, before the introduction of T-Graph [23].

6.1 Discussion

We deployed a focused Web crawler system based on two main hypotheses. The first was to increase the accuracy in prediction of the subject/topic of an unvisited Web page for which we extracted and utilized specific determinant HTML elements from the parent page. Our results showed the effectiveness of this assumption in the dimensions of a prototype. The second hypothesis was to guide the crawler to find the closest-to-optimal path to harvest the Web with the use of a custom hierarchical data structure called T-Graph. The T-Graph guides the crawler to find highly populated regions of the Web where on-topic pages are found. The experimental results satisfied this assumption as well.

To complete the system evaluation and in order to portray where in the literature of focused crawler subject field our proposed algorithm fits, the Treasure-Crawler was compared to other methods of crawling, namely the Context Focused Crawler [2], LSCrawler [4], Relevancy Context Graphs [5], Hybrid Focused Crawler [6], Meta Search Crawler [3], HAWK [7], Intelligent Focused Crawler [9], Modified Naïve Bayes [8] and the OntoCrawler [10]. Table 7 presented this comparison in terms of the employed algorithm for each module to crawl the Web, as well as the performance results of the mentioned systems. In our method, we attempted to keep the textual comparisons minimal, and only relied on simple string functions instead of complicated data mining algorithms. This simplistic way of calculations brought about more realistic and reliable results. We experienced several limitations and restrictions in our implementation, some of which definitely affected the results. To name a few: lack of an ideal parallel or distributed Web server with specific configurations, lack of a high-bandwidth Internet connection with appropriate specifications, lack of a complete list of Dewey entries, and shortcomings of the DMOZ dataset.

It was therefore proposed that any further expansion or implementation of the Treasure-Crawler be carried out by pre-satisfying these constraints. For example, the availability of the complete Dewey system entries will definitely improve the performance of the system in terms of accuracy and preciseness. In our test runs, the DMOZ data set was considered to represent the whole Web. However, the lack of interconnections of Web pages in this data set affected the performance in harvesting its pages from one to another.

Although the main stated assumptions of this research were satisfied, it was obvious that having the above limitations relaxed, we would gain more remarkable results and a highly reliable and flexible innovative method.

7. Conclusion and Future Directions

Although the page content, hierarchy patterns, and anchor texts are satisfactory leads, a focused crawler inevitably needs a multi-level inspection infrastructure to compensate their drawbacks. Unfortunately the current papers overlook the power of such comprehensiveness [24]. Considering these shortcomings, our proposed Treasure-Crawler utilized a significant approach in crawling and indexing Web pages that complied with its predefined topic of interest. The hierarchical structure of the T-Graph guided the crawler to detect and harvest the regions of the Web that embodied a larger population of on-topic pages. The main idea was to detect the topic boundary of an unvisited URL. This was first manually checked to make sure that the HTML parser was correctly fetching the HTML elements of the page. After the necessary textual data in a page was extracted, the system performed one of its major tasks; detecting the topical focus of the unvisited page. The next major task was to assign a score to each URL. This was carried out by use of the pre-constructed but updating T-Graph. Also, seed URLs played a key role in the Treasure-Crawler system. The generality of the generic seeds as well as the relatedness of on-topic seeds were highly important.

Crawling the deep Web as well as tallying with Web 2.0, Web 3.0, and further Web 4.0 concepts is a challenging task; an advancement that goes beyond the content, semantic Web and ontologies by interweaving bits of content into a mix of derived meta-knowledge. The Treasure-Crawler takes advantage of basic

specifications of Web 2.0 when encountering user generated data and in the scope of HTML elements, though there are more structures to handle in its future versions to support semantic and ontology-based crawling. Moreover, organizations' current concern with scalability and agility of their big data predictive analytics has recently pushed them towards usage of customized focused (topic-specific) crawlers as a complement to enhance searchers. As a result, using advanced crawlers *attuned* to semantic Web, built on Web 2.0, and leaping towards Web 4.0 is the way forward. We envisage this as a major possible further improvement in future designs of our proposed system.

A future direction is in the design of a real search engine based on Treasure-Crawler. There will be the need for a distributed infrastructure which synchronizes several Treasure-Crawlers that specialize in diverse topics. Then an index synchronizer will virtually incorporate these repositories. The search engine then establishes and manages appropriate connections from the user interface to this merged index of the Web.

In comparing and evaluating the Treasure-Crawler against other focused crawlers, some factors are necessary to be considered. For example, the semantic crawlers should be evaluated in different terms as they are employing a more sophisticated framework – the semantic Web. Another variable that should be taken into account is the number of crawled pages, which should be equal for all the crawlers in the comparison. More importantly, the crawlers in comparison must focus on a common topic of interest, since the topical communities are absolutely diverse on the Internet. Obviously, if one decides to design a new focused crawler, these diverse factors are of a great help to aim for optimality in crawling.

7.1 Future R&D Directions

Focused crawling is an exciting research domain, where making any change in any parts of the system will affect the overall performance of the system. This changeability and response is what makes it a challenge for the scientists to continue their research to obtain the desired results. The following changes and additions are just some of many issues that can be taken into account for further developments of the Treasure-Crawler:

- *Lemmatization* is the use of a vocabulary as well as morphological analysis of words, the goal of which is to remove endings related to inflection, and to return the base form of a word, also called lemma. Since there are some inefficiencies or errors in measurements with stemming algorithms, such as over-stemming and under-stemming, it is suggested that a lemmatization algorithm may be used in parallel with the stemming algorithm.
- According to the use of different terminology for similar concepts plus the use of combinatory terms (called conflation) as a typical human behavior, many text materials on the Web might be differently written but convey the same meaning. Regarding this fact, using a thesaurus in order to better detect the topical focus of an unvisited link is suggested. While the system is equipped with such resources, the text similarity measurements will take a semi-semantic way becoming more effective and reliable. For this issue, WordNet [25] is a suggested tool that can be furnished into the system resulting in a more semantic nature.
- As a significant problem in the process of natural languages, *word sense disambiguation* aims to determine which sense of the word (when it has several meanings) is meant to be used in a sentence. To overcome this issue, several algorithms as well as lexical resources are introduced and can be used. As the text comparison and topic detection are very important processes in a focused Web crawler, arming the system with these capabilities will pragmatically increase the accuracy of the system.
- Seeing the Web as a large system, in addition to the elements of a Web page, there are several functionalities that can touch the accuracy of classification and prioritization. One of these factors is the users' activity and the frequency of a page being visited. Including the concept of clickstream has already been studied and proven to be effective in Web crawlers [11]. We are currently studying the possibility of adding such functionality to our system as a plug-and-play component.
- One of the best known algorithms in prioritizing the links in a Web page is PageRank which takes into account the incoming and outgoing links to and from the Web page in order to calculate the priority score. It is suggested that PageRank (or a modification) may be embedded into the Treasure-Crawler to increase its performance more efficiently.
- A system facility should be present in such a crawler system to control the different behaviors of the repository which is actually a large database. For example, as already included in the current version of this system, every time a Web page is about to be stored, the DB Controller component checks whether the page is already stored or not. If yes, one alternative is that the record is replaced keeping the older version as history. This facility can basically depend on the concept of triggering in order to be implemented.

- One of the major problems that we solved but needs more improvements is how to connect highly populated regions of the Web that are slightly connected. The concern is how we should guide the crawler to escape the populated region, follow the URLs and get to the next populated region. BDS is one remarkable strategy that is originally designed to connect social networks through loose links but the view point of the authors can provide fruitful directions for our purpose [12].
- Many of the Web developers use commercial and non-commercial design templates which causes the Web pages to have similar HTML DOM trees. This causes a phenomenon called *noise* which possibly affects the classification process. A noise detection and reduction system can provably increase the accuracy of the classification process [26].
- If the approach to the construction of T-Graph is chosen to be bottom-up, then the search API of the Google is suggested to be used. In this case, the back crawling approach conducts the search for the parents of each node of the graph. Taking this approach has several benefits as well as drawbacks. Using the Internet is surely slower and requires additional resources. On the other hand, the efficiency of the graph, in terms of quality of the nodes and links, will be increased since the current search engines such as Google use highly intelligent algorithms in classification of pages and topics.
- Although the function to calculate OSM is defined to be the average of the four similarity values, it is suggested that it assigns a weight to each of those partial similarity values according to their effect. This effect weight should be calculated experimentally to gain the minimum error rate. For example, we can compare parts of all the nodes in a level and then normalize the values by dividing each by the sum of all (ratio). This will be an optimization problem called "constraint optimization", which means setting these weights (of SIMs) to reach the minimum error where the summation of all values will be 1, and each of them is between 0 and 1.
- It is considered that the priority score of an unvisited link is the inverse of the least number of links that should be traversed to reach the target level in the T-Graph. Another suggested solution to calculate the priority in a more effective way is to consider the level number of all the similar nodes as well as their link distance to reach the target level simultaneously. Also we can add the OSM of each node into the formula for a better and more logical outcome.
- Although the proposed system relies mainly on the concept of T-Graph, some significant changes in the utilized data structure can speed up the prioritization process. One of our planned changes is to utilize a data structure with the least possible traversal time complexity such as Trie with $O(n)$.
- In addition to the HTML elements of a Web page, some recent researches have shown the effectiveness of using the concepts of semantic Web and ontologies, as well as going beyond the definition level to the context level with utilization of XML pieces of information [27]. It should be noted that these functionalities should be added to any design with respect to an important point; there should be a balance between the implemented capabilities and the complexity of the system.
- Spam detection and filtering is one important part of a Web crawler to curtail annoyance and misrepresentation of focused and efficient searches. There are different types of spam and distortions, such as repeating keywords, invisible embedded text, automatically generated gibberish, sneaky redirects, illegitimate user-generated content especially on forums, etc. This causes legitimate sites to be buried and not found by the crawler, and motivates the idea of including a key component in the structure of commercial crawlers to detect and handle spam content accordingly.

The above given directions are only some of many ideas that can improve further versions of the Treasure-Crawler. According to the unstoppable growth of the Web, focused crawlers are currently forming an interesting research area, although it is hard work to compete with existing sophisticated search engines.

Finally, the idea of using significant HTML elements of a parent Web page in predicting the subject of its child pages provides flexibility. There are diverse approaches to classify Web pages into specific categories of human knowledge; however researchers may prefer to design custom methods/models as proposed in this paper. The use of a custom designed structure of exemplary documents is one other subject to investigate further. This structure is what differentiates focused crawlers from topical crawlers, and what makes focused crawlers more widely researched on.

References

[1] A. Seyfi and A. Patel, "A focused crawler combinatory link and content model based on T-Graph principles," *Computer Standards & Interfaces*, 2015 (in press), DOI: http://dx.doi.org/10.1016/j.csi.2015.07.001

[2] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori, "Focused crawling using context graphs," in *26th International Conference on Very Large Databases, VLDB*, Cairo, Egypt, 2000, pp. 527-534.

[3] J. Qin, Y. Zhou, and M. Chau, "Building domain-specific Web collections for scientific digital libraries: a meta-search enhanced focused crawling method," in *Joint ACM/IEEE Conference on Digital Libraries*, Tucson, Arizona, USA, 2004.

[4] M. Yuvarani, N. C. S. N. Iyengar, and A. Kannan, "LSCrawler: a framework for an enhanced focused Web crawler based on link semantics," in *IEEE/WIC/ACM International Conference on Web Intelligence*, 2006.

[5] C.-C. Hsu and F. Wu, "Topic-specific crawling on the Web with the measurements of the relevancy context graph," *Information Systems*, vol. 31, pp. 232-246, 2006, DOI:

[6] M. Jamali, H. Sayyadi, B. B. Hariri, and H. Abolhassani, "A method for focused crawling using combination of link structure and content similarity," in *IEEE/WIC/ACM International Conference on Web Intelligence*, 2006.

[7] C. Xiaoyun and Z. Xin, "HAWK: a focused crawler with content and link analysis," in *e-Business Engineering, 2008. ICEBE '08. IEEE International Conference on*, 2008, pp. 677-680.

[8] W. Wang, X. Chen, Y. Zou, H. Wang, and Z. Dai, "A focused crawler based on Naïve Bayes classifier," in *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*, 2010, pp. 517-521.

[9] D. Taylan, M. Poyraz, S. Akyokus, and M. C. Ganiz, "Intelligent focused crawler: learning which links to crawl," in *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, 2011, pp. 504-508.

[10] O. Jalilian and H. Khotanlou, "A new fuzzy-based method to weigh the related concepts in semantic focused web crawlers," in *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, 2011, pp. 23-27.

[11] F. Ahmadi-Abkenari and A. Selamat, "An architecture for a focused trend parallel Web crawler with the application of clickstream analysis," *Information Sciences*, vol. 184, pp. 266-281, 2012, DOI: http://dx.doi.org/10.1016/j.ins.2011.08.022

[12] F. Buccafurri, G. Lax, A. Nocera, and D. Ursino, "Moving from social networks to social internetworking scenarios: the crawling perspective," *Information Sciences*, vol. 256, pp. 126-137, 2014, DOI: http://dx.doi.org/10.1016/j.ins.2013.08.046

[13] M. P. Basgalupp, R. C. Barros, A. C. P. L. F. de-Carvalho, and A. A. Freitas, "Evolving decision trees with beam search-based initialization and lexicographic multi-objective evaluation," *Information Sciences*, vol. 258, pp. 160-181, 2014, DOI: http://dx.doi.org/10.1016/j.ins.2013.07.025

[14] A. Moosavi, S. Hooshmand, S. Baghbanzadeh, G.-V. Jourdan, G. Bochmann, and I. Onut, "Indexing Rich Internet Applications using components-based crawling," in *Web Engineering*. vol. 8541, S. Casteleyn, G. Rossi, and M. Winckler, Eds., ed: Springer International Publishing, 2014, pp. 200-217.

[15] A. van Deursen, A. Mesbah, and A. Nederlof, "Crawl-based analysis of web applications: Prospects and challenges," *Science of Computer Programming*, vol. 97, Part 1, pp. 173-180, 1/1/ 2015, DOI: http://dx.doi.org/10.1016/j.scico.2014.09.005

[16] Frank. (30 June 2015). *Let's Do Dewey*. Available: http://library.mtsu.edu/dewey

[17] OCLC. (30 June 2015). *Dewey Services at a glance*. Available: http://www.oclc.org/dewey/

[18] MOZ. (30 June 2015). *Open Site Explorer*. Available: http://moz.com/researchtools/ose

[19] DMOZ. (1998, 30 June 2015). *Open Directory Project*. Available: http://www.dmoz.org

[20] F. Menczer, G. Pant, P. Srinivasan, and M. E. Ruiz, "Evaluating topic-driven Web crawlers," in *SIGIR'01*, New Orleans, Louisiana, USA, 2001.

[21] S. Chakrabarti, M. v. d. Berg, and B. Domc, "Focused crawling: a new approach to topic-specific Web," in *The 8th World-Wide Web conference (WWW8)*, 1999.

[22] P. Srinivasan, F. Menczer, and G. Pant, "A general evaluation framework for topical crawlers," *Information Retrieval*, vol. 8, pp. 417-447, 2005, DOI: 10.1007/s10791-005-6993-5

[23] A. Patel, "An adaptive updating topic specific Web search system using T-Graph," *Journal of Computer Science*, vol. 6, pp. 450-456, 2010, DOI: http://dx.doi.org/10.3844/jcssp.2010.450.456

[24] R. Chen, "An enhanced Web robot for the Cindi system," 2008.

[25] U. Princeton. (30 June 2015). *WordNet*. Available: http://wordnet.princeton.edu/wordnet/

[26] D. Alassi and R. Alhajj, "Effectiveness of template detection on noise reduction and websites summarization," *Information Sciences*, vol. 219, pp. 41-72, 2013, DOI: http://dx.doi.org/10.1016/j.ins.2012.07.022

[27] P. D. Hossein-Zadeh and M. Z. Reformat, "Assessment of semantic similarity of concepts defined in ontology," *Information Sciences*, vol. 250, pp. 21-39, 2013, DOI: http://dx.doi.org/10.1016/j.ins.2013.06.056

Research Highlights:

- We present the experimental results of a focused Web crawler that combines link-based and content-based approaches to predict the topical focus of an unvisited page
- We present a custom method using Dewey Decimal Classification system to best classify the subject of an unvisited page into standard human knowledge categories
- To prioritize an unvisited URL, we use a dynamic, flexible and updating hierarchical data structure called T-Graph that helps find the shortest path to get to on-topic pages on the Web
- For the background review, the experimental results from several crawlers are presented
- We compare our results against other significant focused Web crawlers