International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 05 | Aug-2015 www.irjet.net p-ISSN: 2395-0072

WEB FORUMS CRAWLER FOR ANALYSIS OF USER SENTIMENTS

Dr.D.Devakumari[1], R.Komalavalli[2]

[1] *Assistant Professor, PG and Research Department of Computer Science, Government Arts College(Autonomous), Coimbatore, Tamil Nadu, India.*
[2] *Research Scholar ,Department of Computer Science , L.R.G Government Arts College For Women , Tirupur, Tamil Nadu, India.*

Abstract: Forum Crawler Under Supervision (FoCUS), is a supervised web-scale forum crawler. The goal of FoCUS is to crawl relevant forum content from the web with minimal overhead. Forum threads contain information content that is the target of forum crawlers. Although forums have different layouts or styles and are powered by different forum software packages, they always have similar implicit navigation paths connected by specific URL types to lead users from entry pages to thread pages. Based on this observation, we reduce the web forum crawling problem to a URL-type recognition problem. And we show how to learn accurate and effective regular expression patterns of implicit navigation paths from automatically created training sets using aggregated results from weak page type classifiers. Robust page type classifiers can be trained from as few as five annotated forums and applied to a large set of unseen forums. Our test results show that FoCUS achieved over 98 percent effectiveness and 97 percent coverage on a large set of test forums powered by over 150 different forum software packages. In addition, the results of applying FoCUS on more than 100 community Question and Answer sites and Blog sites demonstrated that the concept of implicit navigation path could apply to other social media sites.

Key Words: *EIT path, forum crawling, ITF regex, page classification, page type, URL pattern learning, URL type*

1. INTRODUCTION

INTERNET forums [4] (also called web forums) are important services where users can request and exchange information with others. For example, the TripAdvisor Travel Board is a place where people can ask and share travel tips. Due to the richness of information in forums, researchers are increasingly interested in mining knowledge from them. Zhai and Liu [28], Yang et al. [27], and Song et al. [23] extracted structured data from forums. Gao et al. [15] identified question and answer pairs in forum threads. Zhang et al. [30] proposed methods to extract and rank product features for opinion mining from forum posts. Glance et al. [16] tried to mine business intelligence from forum data. Zhang et al. [29] proposed algorithms to extract expertise network in forums. To harvest knowledge from forums, their content must be downloaded first. However, forum crawling is not a trivial problem. Generic crawlers [12], which adopt a breadth-first traversal strategy, are usually ineffective and inefficient for forum crawling. This is mainly due to two non crawler friendly characteristics of forums [13], [26]: 1) duplicate links and uninformative pages and 2) page-flipping links. A forum typically has many duplicate links that point to a common page but with different URLs [7], e.g., shortcut links pointing to the latest posts or URLs for user experience functions such as “view by date” or “view by title.” A generic crawler that blindly follows these links will crawl many duplicate pages, making it inefficient. A forum also has many uninformative pages such as login control to protect user privacy or forum software specific FAQs. Following these links, a crawler will crawl many uninformative pages. Though there are standard-based methods such as specifying the “rel” attribute with the “nofollow” value (i.e., “rel ¼ nofollow”) [6], Robots Exclusion Standard (robots.txt) [10], and Sitemap [9] [22] for forum operators to instruct web crawlers on how to crawl a site effectively, we found that over a set of nine test forums more than 47 percent of the pages crawled by a breadth-first crawler following these protocols were duplicates or uninformative. This number is a little higher than the 40 percent that Cai et al. [13]

reported but both show the inefficiency of generic crawlers. More information about this testing can be found in Section 5.2.1. Besides duplicate links and uninformative pages, a long forum board or thread is usually divided into multiple pages which are linked by page-flipping links, for example,
see Figs. 2, 3b, and 3c. Generic crawlers process each page individually and ignore the relationships between such pages. These relationships should be preserved while crawling to facilitate downstream tasks such as page wrapping and content indexing [27]. For example, multiple pages belonging to a thread should be concatenated together in order to extract all the posts in the thread as well as the reply-relationships between posts. In addition to the above two challenges, there is also a problem of entry URL discovery. The entry URL of a forum points to its homepage, which is the lowest common ancestor page of all its threads. Our experiment "Evaluation of Starting from Non-Entry URLs" shows that a crawler starting from an entry URL can achieve a much higher performance than starting from nonentry URLs. Previous works by Vidal et al. [25] and Cai et al. [13] assumed that an entry URL is given.

2. RELATED WORK

Vidal et al. [25] proposed a method for learning regular expression patterns of URLs that lead a crawler from an entry page to target pages. Target pages were found through comparing DOM trees of pages with a preselected sample target page. It is very effective but it only works for the specific site from which the sample page is drawn. The same process has to be repeated every time for a new site. Therefore, it is not suitable for large-scale crawling. In contrast, FoCUS learns URL patterns across multiple sites and automatically finds a forum's entry page given a page from the forum. Experimental results show that FoCUS is effective at large-scale forum crawling by leveraging crawling knowledge learned from a few annotated forum sites.
Guo et al. [17] and Li et al. [20] are similar to our work.
However, Guo et al. did not mention how to discover and traverse URLs. Li et al. developed some heuristic rules to discovery URLs. However, their rules are too specific and can only be applied to specific forums powered by the particular software package in which the heuristics were conceived. Unfortunately, according to ForumMatrix [2], there is hundreds of different forum software packages used on the Internet. Please refer to [2], [3], [5] for more information about forum software packages. In addition, many forums use their own customized software. A recent and more comprehensive work on forum crawling is iRobot by Cai et al. [13]. iRobot aims to automatically learn a forum crawler with minimum human intervention by sampling pages, clustering them, selecting informative clusters via an informativeness measure, and finding a traversal path by a spanning tree algorithm. However, the traversal path selection procedure requires human inspection. Follow up work by Wang et al. [26] proposed an algorithm to address the traversal path selection problem. They introduced the concept of skeleton link and page-flipping link. Skeleton links are "the most important links supporting the structure of a forum site." Importance is determined by informativeness and coverage metrics. Page-flipping links are determined using connectivity metric. By identifying and only following skeleton links and page-flipping links, they showed that iRobot can achieve effectiveness and coverage. According to our evaluation, its sampling strategy and informativeness estimation is not robust and its tree-like traversal path does not allow more than one path from a starting page node to a same ending page node. For example, there are six paths from entry to threads. But iRobot would only take the first path (entry ! board ! thread). iRobot learns URL location information to discover new URLs in crawling, but a URL location might become invalid when the page structure changes. As opposed to iRobot, we explicitly define entry-index-thread paths and leverage page layouts to identify index pages and thread pages. FoCUS also learns URL patterns instead of URL locations to
discover new URLs. Thus, it does not need to classify new pages in crawling and would not be affected by a change in page structures. The respective results from iRobot and FoCUS demonstrated that the EIT paths and URL patterns are more robust than the traversal path and URL location feature in iRobot.

Another related work is near-duplicate detection. Forum crawling also needs to remove duplicates. But contentbased duplicate detection [18], [21] is not bandwidthefficient, because it can only be carried out when pages have been downloaded. URL-based duplicate detection [14], [19] is not helpful. It tries to mine rules of different URLs with similar text. However, such methods still need to analyze logs from sites or results of a

previous crawl. In forums, index URLs, thread URLs, and page-flipping URLs have specific URL patterns. Thus, in this paper, by learning patterns of index URLs, thread URLs, and page-flipping URLs and adopting a simple URL string de-duplication technique (e.g., a string hashset), FoCUS can avoid duplicates without duplicate detection. To alleviate unnecessary crawling, industry standards such as "nofollow" [6], Robots Exclusion Standard (robots. txt) [10], and Sitemap Protocol [9], [22] have been introduced. By specifying the "rel" attribute with the "nofollow" value (i.e., "rel ¼ nofollow"), page authors can inform a crawler that the destination content is not endorsed. However, it is intended to reduce the effectiveness of search engine spams, but not meant for blocking access to pages. A proper way is robots.txt [10]. It is designed to specify what pages a crawler is allowed to visit or not. Sitemap [9] is an XML file that lists URLs along with additional metadata including update time, change frequency etc. Generally speaking, the purpose of robots.txt and Sitemap is to enable the site to be crawled intelligently. So they may be useful to forum crawling. However, it is difficult to maintain such files for forums as their content continually changes. In our experiment more than 47 percent of the pages crawled by a generic crawler which can properly understand these industry standards are uninformative or duplicates.

3. METHODS

To learn ITF regexes, FoCUS adopts a two-step supervised training procedure. The first step is training sets construction. The second step is regexes learning.

3.1. Constructing URLTraining Sets

The goal of URL training sets construction is to automatically create sets of highly precise index URL, thread URL, and page-flipping URL strings for ITF regexes learning. Its use a similar procedure to construct index URL and thread URL training sets since they have very similar properties except for the types of their destination pages; to present this part first. Page-flipping URLs have their own specific properties that are different from index URLs and thread URLs; we present this part later.

3.2. Index URL and Thread URL Training Sets

Recall that an index URL is a URL that is on an entry or index page; its destination page is another index page; its anchor text is the board title of its destination page. A thread URL is a URL that is on an index page; its destination page is a thread page; its anchor text is the thread title of its destination page. It also note that the only way to distinguish index URLs from thread URLs is the type of their destination pages. Therefore, we need a method to decide the page type of a destination page.

The index pages and thread pages each have their own typical layouts. Usually, an index page has many narrow records, relatively long anchor text, and short plain text; while a thread page has a few large records (user posts). Each post has a very long text block and relatively short anchor text.

An index page or a thread page always has a timestamp field in each record, but the timestamp order in the two types of pages are reversed: the timestamps are typically in descending order in an index page while they are in ascending order in a thread page. In addition, each record in an index page or a thread page usually has a link pointing to a user profile page.

3.3. Page Flipping URL Training Set

Page-flipping URLs point to index pages or thread pages but they are very different from index URLs or thread URLs. The proposed "connectivity" metric is used to distinguish page-flipping URLs from other loop-back URLs. However, the metric only works well on the "grouped" page-flipping URLs, i.e., more than one page-flipping URL in one page.

But in many forums, there is only one page-flipping URL in one page, which we called single page-flipping URL. Such URLs cannot be detected using the "connectivity" metric. To address this shortcoming, we observed some special properties of page flipping URLs and proposed an algorithm to detect page flipping URLs based on these properties.

In particular, the grouped page-flipping URLs have the following properties:

1. Their anchor text is either a sequence of digits such as 1, 2, 3, or special text such as "last."

2. They appear at the same location on the DOM tree of their source page and the DOM trees of their destination pages.

3. Their destination pages have similar layout with their source pages. We use tree similarity to determine whether the layouts of two pages are similar or not. As to single page-flipping URLs, they do not have the property 1, but they have another special property.

4. The single page-flipping URLs appearing in their source pages and their destination pages have the same anchor text but different URL strings.

3.4. K-Means Clustering Algorithm

The non-hierarchical method initially takes the number of components of the population equal to the final required number of clusters. First, the final required number of clusters is chosen such that the points are mutually farthest apart. Next, it examines each component in the population and assigns it to one of the clusters depending on the minimum distance.

The centroid's position is recalculated every time a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters.

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result.

So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done.

At this point they need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After they have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop they may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

The k-means approach to clustering performs an iterative alternating fitting process to form the number of specified clusters. The k-means method first selects a set of n points called cluster seeds as a first guess of the means of the clusters. Each observation is assigned to the nearest seed to form a set of temporary clusters. The seeds are then replaced by the cluster means, the points are reassigned, and the process continues until no further changes occur in the clusters.

The Algorithm is as follows

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the K centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

The K-Means Algorithm Process

1. The dataset is partitioned into K clusters and the data points are randomly assigned to the clusters resulting in clusters that have roughly the same number of data points.

2. For each data point:

3. Calculate the distance from the data point to each cluster.

4. If the data point is closest to its own cluster, leave it where it is. If the data point is not closest to its own cluster, move it into the closest cluster.

5. Repeat the above step until a complete pass through all the data points results in no data point moving from one cluster to another. At this point the clusters are stable and the clustering process ends.

6. The choice of initial partition can greatly affect the final clusters that result, in terms of inter-cluster and intracluster distances and cohesion.

4. EXPERIMENTAL RESULTS

The following **Table 5.1** describes experimental result for proposed system for downloading the positive command details. The table contains forum id and corresponding average number of positive details are shown.

Table 5.1 Positive Forum Command Analysis (Count)

S.NO	FORUM ID	POSITIVE PERCENT
1	1	486
2	2	5036
3	3	3832
4	4	2180
5	5	1552
6	6	4696
7	7	3796
8	8	1824
9	9	2012
10	10	3320
11	11	4616
12	12	2410
13	13	2322
14	14	2286
15	15	2676
16	16	2742
17	17	1959
18	18	1662
19	19	3918
20	20	1904

The proposed methodology efficiently analyzes their sentiments. An incomparable advantage of the proposed model is that it easily scales to handle networks with millions of posts. Since the proposed model is sensitive to the number of social dimensions as shown in the experiment, further research is needed to determine a suitable dimensionality automatically.

The following **Table 5.2** describes experimental result for proposed system for downloading the negative command analysis details. The table contains forum id and corresponding average number of negative command details are shown.

Table 5.2 Negative Forum Command Analysis (Count)

S.NO	FORUM ID	NEGATIVE PERCENT
1	1	18
2	2	4
3	3	0
4	4	0
5	5	0
6	6	0
7	7	3
8	8	6
9	9	3
10	10	0
11	11	3
12	12	0
13	13	3
14	14	0
15	15	15
16	16	6

17	17	6
18	18	6
19	19	6
20	20	0

The following **Fig 5.1** describes experimental result for proposed system for downloading the positive command details. The figures contains forum id and corresponding average number of positive details are shown.

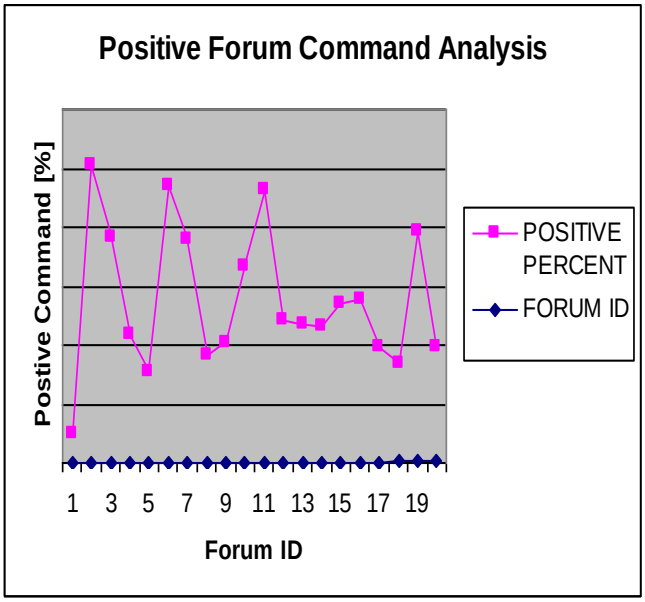

Fig 5.1 Positive Forum Command Analysis(count)

The following **Fig 5.2** describes experimental result for proposed system for downloading the negative command analysis details. The figures contains forum id and corresponding average number of negative command details are shown.

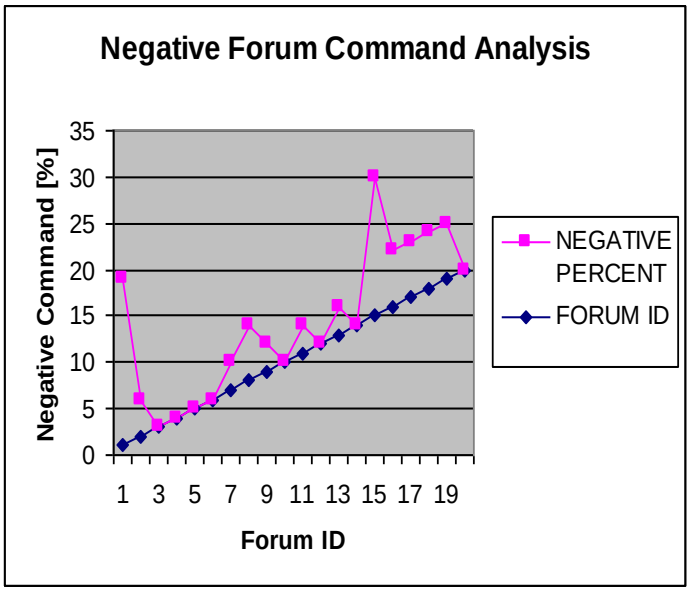

Fig 5.2 Negative Forum Command Analysis (Count)

Table 5.3 Analyzing average post per forum and average sentimental value

Forum Id	Forum Title	Post Count	Avg Post Per forum	Avg sentiment value per forum
1	Google	1340	335	0
34	Google+	1158	22	1
37	Digital Point Ads	708	14	1
38	Google AdWords	684	12	0
39	Yahoo Search Marketing	1240	24	1
44	Google	2094	41	0
46	Azoogle	1516	29	0
49	ClickBank	1352	27	0
52	General Business	1206	23	0
54	Payment Processing	1782	34	0
59	Copywriting	526	10	0
62	Sites	504	9	1
63	Domains	78	1	1
66	eBooks	484	9	1
70	Content Creation	206	4	1

71	Design	498	9	1
72	Programming	202	3	1
77	Template Sponsorship	94	2	1
82	Adult	30	0	1
83	Design & Development	0	0	1
84	HTML & Website Design	254	4	1
85	CSS	110	2	1
86	Graphics & Multimedia	79	1	0

Note: Avg - Average

The following **Fig 5.3** describes the graphical representation of analyzing average post forum and average sentimental value.

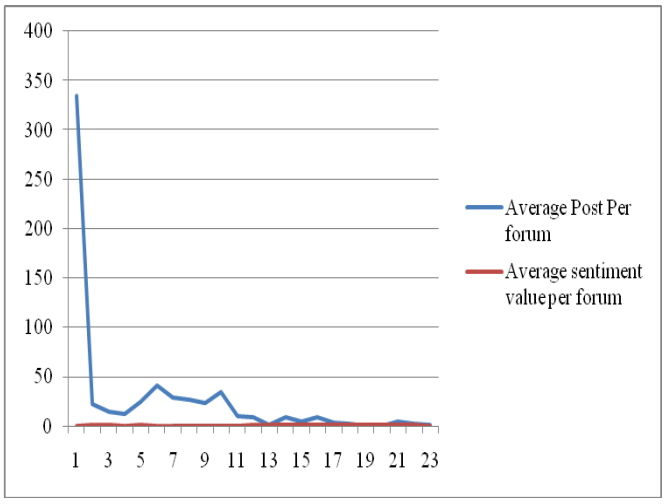

Fig5.3 Analyzing Average Post Per Forum And Average Sentimental Value

The proposed approach includes group the forums into various clusters using emotional polarity computation and integrated sentiment analysis based on K-means clustering. Also positive and negative replies are clustered. Using scalable learning the relationship among the topics are identified and represent it as a graph. Data are collected from forums.digitalpoint.com which includes a range of 75 different topic forums. Computation indicates that within the same time window, forecasting achieves highly consistent results with K-means clustering.

Also the forum topics are represented using graphs. In this graph the is used to represent the forum titles, thread count, post count, average post per forum, average sentiment value per forum and the similarity or relationship between the topics.

5. CONCLUSION

In this thesis, the algorithms are developed to automatically analyze the emotional polarity of a text, based on which a value for each piece of text is obtained. The absolute value of the text represents the influential power and the sign of the text denotes its emotional polarity.

This K-means clustering is applied to develop integrated approach for online sports forums cluster analysis. Clustering algorithm is applied to group the forums into various clusters, with the center of each cluster representing a hotspot forum within the current time span.

In *addition* to clustering the forums based on data from the current time window, it is also conducted forecast for the next time window. Empirical studies present strong proof of the existence of correlations between post text sentiment and hotspot distribution. Education Institutions, as information seekers can benefit from the hotspot predicting approaches in several ways. They should follow the same rules as the academic objectives, and be measurable, quantifiable, and time specific. However, in practice parents and students behavior are always hard to be explored and captured.

Using the hotspot predicting approaches can help the education institutions understand what their specific customers' timely concerns regarding goods and services information. Results generated from the approach can be also combined to competitor analysis to yield comprehensive decision support information.

6. FUTURE ENHANCEMENT

The future, how to utilize the inferred information and extend the framework for efficient and effective network monitoring and application design

The new system become useful if the below enhancements are made in future.

- The application can be web service oriented so that it can be further developed in any platform.
- The application if developed as web site can be used from anywhere.
- At present, number of posts/forum, average sentiment values/forums, positive % of posts/forum and negative % of posts/forums are taken as feature spaces for K-Means clustering. In future, neutral replies, multiple-languages based replies can also be taken as dimensions for clustering purpose.
- In addition, currently forums are taken for hot spot detection. Live Text streams such as chatting messages can be tracked and classification can be adopted.

The new system is designed such that those enhancements can be integrated with current modules easily with less integration work. The new system becomes useful if the above enhancements are made in future. The new system is designed such that those enhancements can be integrated with current modules easily with less integration work.

REFERENCES

[1] Blog, http://en.wikipedia.org/wiki/Blog, 2012.
[2]"ForumMatrix,"http://www.forummatrix.org/index.php, 2012.
[3] Hot Scripts,
http://www.hotscripts.com/index.php, 2012.
[4]InternetForum,http://en.wikipedia.org/wiki/Internet_forum,
[5] "Message Boards Statistics," http://www.big-boards.com/statistics/, 2012.
[6] nofollow, http://en.wikipedia.org/wiki/Nofollow, 2012.
[7] "RFC 1738—Uniform Resource Locators (URL)," http://www.ietf.org/rfc/rfc1738.txt, 2012.
[8] Session ID,
http://en.wikipedia.org/wiki/Session_ID, 2012.
[9]"TheSitemapProtocol,"http://sitemaps.org/protocol.php, 2012.
[10]"TheWeb Robots Pages,"
http://www.robotstxt.org/, 2012.
[11] "WeblogMatrix,"
http://www.weblogmatrix.org/, 2012.
[12] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine." Computer Networks and ISDN Systems, vol. 30, nos. 1-7, pp. 107-117, 1998.
[13] R. Cai, J.-M. Yang, W. Lai, Y. Wang, and L. Zhang, "iRobot: An Intelligent Crawler for Web Forums," Proc. 17th Int'l Conf. World Wide Web, pp. 447-456, 2008.
[14] A. Dasgupta, R. Kumar, and A. Sasturkar, "De-Duping URLs via Rewrite Rules," Proc. 14th ACM SIGKDD Int'l Conf. Knowledge
Discovery and Data Mining, pp. 186-194, 2008.
[15] C. Gao, L. Wang, C.-Y. Lin, and Y.-I. Song, "Finding Question- Answer Pairs from Online Forums," Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 467-474, 2008.
[16] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, "Deriving Marketing Intelligence from Online Discussion," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 419-428, 2005.
[17] Y. Guo, K. Li, K. Zhang, and G. Zhang, "Board Forum Crawling: A Web Crawling Method for Web Forum," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence, pp. 475-478, 2006.
[18] M. Henzinger, "Finding Near-Duplicate Web Pages: A Large- Scale Evaluation of Algorithms," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 284-291, 2006.
[19] H.S. Koppula, K.P. Leela, A. Agarwal, K.P. Chitrapura, S. Garg, and A. Sasturkar, "Learning URL Patterns for Webpage De- Duplication," Proc. Third ACM Conf. Web Search and Data Mining, pp. 381-390, 2010.
[20] K. Li, X.Q. Cheng, Y. Guo, and K. hang,"Crawling Dynamic Web Pages in WWW Forums," Computer Eng., vol. 33, no. 6, pp. 80-82, 2007.
[21] G.S. Manku, A. Jain, and A.D. Sarma, "Detecting Near-Duplicates for Web Crawling," Proc. 16th Int'l Conf. World Wide Web, pp. 141- 150, 2007.
[22] U. Schonfeld and N. Shivakumar, "Sitemaps: Above and Beyond the Crawl of Duty," Proc. 18th Int'l Conf. World Wide Web, pp. 991- 1000, 2009.
[23] X.Y. Song, J. Liu, Y.B. Cao, and C.-Y. Lin, "Automatic Extraction of Web Data Records Containing User-Generated Content," Proc. 19th Int'l

Conf. Information and Knowledge Management, pp. 39-48,2010.
[24] V.N. Vapnik, The Nature of Statistical Learning Theory. Springer, 1995.
[25] M.L.A. Vidal, A.S. Silva, E.S. Moura, and J.M.B. Cavalcanti, "Structure-Driven Crawler Generation by Example," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 292-299, 2006.
[26] Y. Wang, J.-M. Yang, W. Lai, R. Cai, L. Zhang, and W.-Y. Ma,"Exploring Traversal Strategy for Web Forum Crawling," Proc.31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 459-466, 2008.

BIOGRAPHIES

Dr. D. Devakumari has received M. Phil degree from Manonmaniam Sundaranar University in 2003 and Ph.D from Mother Teresa Womens' University in 2013. Currently she is working as Assistant Professor in the PG and Research Department of Computer Science, Government Arts College (Autonomous), Coimbatore, India. Her research papers have been published in International journals including Inderscience, Springer etc. She has presented papers in National and International Conferences. Her research interests include Data Pre-processing and Pattern Recognition.

Ms. R.Komalavalli has received B.SC(CS) degree from Maharaja Arts and Science College and M.SC(IT) from Maharaja Arts and Science College. Pursuing her M.Phil degree from L.R.G Government Arts College for Women. Currently she is working as Assistant Professor in Department of Computer Science, L.R.G Government Arts College for Women, Tirupur, India.