

WEB FORUMS CRAWLER FOR ANALYSIS USER SENTIMENTS

^IB.Nithya M.sc., ^{II}K. Devika M.Sc., MCA., M.Phil.,

^I Research Scholar, Bharathiar University, Coimbatore, ^{II} Assistant Professor, CS,
^{I,II}Dept. of Computer Science, Maharaja Co-Education College of Arts and Science,
Perundurai, Erode – 638052.

^I Email id: bmsnithya2@gmail.com

^I Contact No:9965112440

^{II} Email id: devika_tarun@yahoo.co.in

^I Contact No: 9894831174

ABSTRACT

The advancement in computing and communication technologies enables people to get together and share information in innovative ways. Social networking sites empower people of different ages and backgrounds with new forms of collaboration, communication, and collective intelligence. This project presents Forum Crawler Under Supervision (FoCUS), 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, the web forum crawling problem is reduced to a URL-type recognition problem and classifies them as Index Page, Thread Page and Page-Flipping page. In addition, this project studies how networks in social media can help predict some human behaviors and individual preferences

Keywords: content based retrieval, multimedia databases, search problems.

1. INTRODUCTION

1.1. Data Mining

Data mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Data mining tools predict behaviors and future trends, allowing businesses to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Data mining derives its name from the similarities between searching for valuable information in a large database and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find where the value resides.

Although data mining is still in its infancy, companies in a wide range of industries - including retail, finance, health care, manufacturing transportation, and aerospace - are already using data mining tools and techniques to take advantage of historical data. By using pattern recognition technologies and statistical and mathematical techniques to sift through warehoused information, data mining helps analysts recognize significant facts, relationships, trends, patterns, exceptions and anomalies that might otherwise go unnoticed.

For businesses, data mining is used to discover patterns and relationships in the data in order to help make better business decisions.

Data mining can help spot sales trends, develop smarter marketing campaigns, and accurately predict customer loyalty.

1.2.WEB CRAWLER

A Web crawler is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web indexing. A Web crawler may also be called a Web spider, an ant, an automatic indexer, or (in the FOAF software context) a Web scutter. Web search engines and some other sites use Web crawling or spidering software to update their web content or indexes of others sites' web content. Web crawlers can copy all the pages they visit for later processing by a search engine that indexes the downloaded pages so that users can search them much more quickly. Crawlers can validate hyperlinks and HTML code. They can also be used for web scraping.

WebCrawler was originally a separate search engine with its own database, and displayed advertising results in separate areas of the page. More recently it has been repositioned as a metasearch engine, providing a composite of separately identified sponsored and non-sponsored search results from most of the popular search engines.

A Web crawler starts with a list of URLs to visit, called the *seeds*. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the *crawl frontier*. URLs from the frontier are recursively visited according to a set of policies. The large volume implies that the crawler can only download a limited number of the Web pages within a given time, so it needs to prioritize its downloads. The high rate of change implies that the pages might have already been updated or even deleted.

The number of possible crawlable URLs being generated by server-side software has also made it difficult for web crawlers to avoid retrieving duplicate content. Endless combinations of HTTP GET (URL-based) parameters exist, of which only a small selection will actually return unique content. For example, a simple online photo gallery may offer three options to users, as specified through HTTP GET parameters in the URL.

If there exist four ways to sort images, three choices of thumbnail size, two file formats, and an option to disable user-provided content, then the same set of content can be accessed with 48 different URLs, all of which may be linked on the site. This mathematical combination creates a problem for crawlers, as they must sort through endless combinations of relatively minor scripted changes in order to retrieve unique content.

"Given that the bandwidth for conducting crawls is neither infinite nor free, it is becoming essential to crawl the Web in not only a scalable, but efficient way, if some reasonable measure of quality or freshness is to be maintained." A crawler must carefully choose at each step which pages to visit next.

1.3. COLLECTIVE BEHAVIOR

Collective behavior refers to the behaviors of individuals in a social networking environment, but it is not simply the aggregation of individual behaviors. In a connected environment, individuals' behaviors tend to be interdependent, influenced by the behavior of friends. This naturally leads to behavior correlation between connected users. Take marketing as an example: if our friends buy something, there is a better-than-average chance that we will buy it, too.

This behavior correlation can also be explained by *homophily*. Homophily is a term coined in the 1950s to explain our tendency to link with one another in ways that confirm, rather than test, our core beliefs. Essentially, we are more likely to connect to others who share certain similarities with us. This phenomenon has been observed not only in the many processes of a physical world, but also in online systems. Homophily results in behavior correlations between connected friends.

In other words, friends in a social network tend to behave similarly. The recent boom of social media enables us to study collective behavior on a large scale. Here, behaviors include a broad range of actions: joining a group, connecting to a person, clicking on an ad, becoming

interested in certain topics, dating people of a certain type, etc. In this work, we attempt to leverage the behavior correlation presented in a social network in order to predict collective behavior in social media. Given a network with the behavioral information of some actors, how can we infer the behavioral outcome of the remaining actors within the same network.

It can also be considered as a special case of semi-supervised learning or relational learning where objects are connected within a network. Some of these methods, if applied directly to social media, yield only limited success. This is because connections in social media are rather noisy and heterogeneous. In the next section, we will discuss the connection heterogeneity in social media, review the concept of social dimension, and anatomize the scalability limitations of the earlier model proposed which provides a compelling motivation for this work.

2. PROBLEM FORMULATION

2.1. PROBLEM FORMULATION

To harvest knowledge from forums, their content must be downloaded first. However, forum crawling is not a trivial problem. Generic crawlers, 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.

- 1) Duplicate links and uninformative pages and
- 2) page-flipping links.

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. The system reduces the forum crawling problem to a URL type recognition problem and implement a crawler, FoCUS, to demonstrate its applicability. It shows how to automatically learn regular expression patterns (ITF regexes) that recognize the index URL, thread URL, and page-flipping URL using the page classifiers built from as few as five annotated forums.

To predict collective behavior in social media is being done by understanding how individuals behave in a social networking environment.. In particular, given information about some individuals, how can infer the behavior of unobserved individuals in the same network? A social-dimension-based approach has been shown effective in addressing the heterogeneity of connections presented in social media.

However, the networks in social media are normally of colossal size, involving hundreds of thousands of actors. The scale of these networks entails scalable learning of models for collective behavior prediction. To address the scalability issue, an edge-centric clustering scheme is required to extract sparse social dimensions.

Hence the thesis is proposed. With sparse social dimensions, the project can efficiently handle networks of millions of actors while demonstrating a comparable prediction performance to other non-scalable methods.

While fuzzy c-means is a popular soft-clustering method, its effectiveness is largely limited to spherical clusters. By applying kernel tricks, the kernel fuzzy c-means algorithm attempts to address this problem by mapping data with nonlinear relationships to appropriate feature spaces. Kernel combination, or selection, is crucial for effective kernel clustering.

Unfortunately, for most applications, it is uneasy to find the right combination. At present, there is a risk in clustering images with more noise pixels. Since the image is not clustered well, the existing system is somewhat less efficient.

The problem is aggravated for many real-world clustering applications, in which there are multiple potentially useful cues. For such applications, to apply kernel-based clustering, it is often necessary to aggregate features from different sources into a single aggregated feature.

2.2. OBJECTIVES OF THE RESEARCH

The development in computing and communication technologies enables people to get together and share information in innovative ways. Social networking sites (a recent phenomenon) empower people of different ages and backgrounds with new forms of collaboration, communication, and collective intelligence. This thesis presents Forum Crawler under Supervision (FoCUS), 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, the web forum crawling problem is reduced to a URL-type recognition problem and classifies them as Index Page, Thread Page and Page-Flipping page. In addition, this thesis studies how networks in social media can help predict some human behaviors and individual preferences. In particular, given the behavior of some individuals in a network, how can infer the behavior of other individuals in the same social network? This study can help better understand behavioral patterns of users in social media for applications like social advertising and recommendation.

This study of collective behavior is to understand how individuals behave in a social networking environment. Oceans of data generated by social media like Facebook, Twitter, and YouTube present opportunities and challenges to study collective behavior on a large scale. This thesis aims to learn to predict collective behavior in social media. A social-dimension-based approach has been shown effective in addressing the heterogeneity of connections presented in social media. However, the networks in social media are normally of colossal size, involving hundreds of thousands of actors. The scale of these networks entails scalable learning of models for collective behavior prediction.

To address the scalability issue, the thesis proposes an edge-centric clustering scheme to extract sparse social dimensions. With sparse social dimensions, the proposed approach can efficiently handle networks of millions of actors while demonstrating a comparable prediction performance to other non-scalable methods.

In addition, the thesis includes a new concept called sentiment analysis. Since many automated prediction methods exist for extracting patterns from sample cases, these patterns can be used to classify new cases. The proposed system contains the method to transform these cases into a standard model of features and classes.

METHODOLOGY

4.1 TERMINOLOGY

To facilitate presentation in the following sections, the first define some terms used in this dissertation.

4.1.1 PAGE TYPE

It classified forum pages into page types.

→Entry Page:

The homepage of a forum is contains a list of boards and is also the lowest common ancestor of all threads.

→Index Page:

A page of a board in a forum, which usually contains a table-like structure; each row in it contains information of a board or a thread.

→Thread Page:

A page of a thread in a forum that contains a list of posts with user generated content belonging to the same discussion.

→Other Page:

A page that is not an entry page, index page, or thread page.

4.1.2 URL TYPE

There are four types of URL.

→Index URL:

A URL is on an entry page or index page and points to an index page. Its anchor text shows the title of its destination board.

→Thread URL:

A URL is on an index page and points to a thread page. Its anchor text is the title of its destination thread.

→Page-flipping URL:

A URL leads users to another page of the same board or the same thread. Correctly dealing with page-flipping URLs enables a crawler to download all threads in a large board or all posts in a long thread.

→Other URL:

A URL that is not an index URL, thread URL, or page-flipping URL.

4.1.3 EIT Path:

An entry-index-thread path is a navigation path from an entry page through a sequence of index pages (via index URLs and index page-flipping URLs) to thread pages (via thread URLs and thread page-flipping URLs).

4.1.4 ITF Regex:

An index-thread-page-flipping regex is a regular expression that can be used to recognize index, thread, or page-flipping URLs. ITF regex is what FoCUS aims to learn and applies directly in online crawling. The learned ITF regexes are site specific, and there are four ITF regexes in a site: one for recognizing index URLs, one for thread URLs, one for index page-flipping URLs, and one for thread page-flipping URLs. A perfect crawler starts from a forum entry URL and only follows URLs that match ITF regexes to crawl all forum threads. The paths that it traverses are EIT paths.

4.2. ARCHITECTURE OF FOCUS

The overall architecture of FoCUS as follows. It consists of two major parts: the learning part and the online crawling part. The learning part first learns ITF regexes of a given forum from automatically constructed URL training examples. The online crawling part then applies learned ITF regexes to crawl all threads efficiently. Given any page of a forum, FoCUS first finds its entry URL using the Entry URL Discovery module.

Then, it uses the Index/Thread URL Detection module to detect index URLs and thread URLs on the entry page; the detected index URLs and thread URLs are saved to the URL training sets. Next, the destination pages of the detected index URLs are fed into this module again to detect more index and thread URLs until no more index URL is detected.

After that, the Page-Flipping URL Detection module tries to find page flipping URLs from both index pages and thread pages and saves them to the training sets. Finally, the ITF Regexes Learning module learns a set of ITF regexes from the URL training sets.

Once the learning is finished, FoCUS performs online crawling as follows: starting from the entry URL, FoCUS follows all URLs matched with any learned ITF regex. FoCUS continues to crawl until no page could be retrieved or other condition is satisfied.

4.2.1. ITF REGEXES LEARNING

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.

i. Constructing URL Training 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; to present this part later.

ii. 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, to 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.

Algorithm IndexUrlAndThreadUrlDetection

Input: sp : an entry page or index page

Output: it_group : a group of index/thread URLs

```
1: let  $it\_group$  be  $\varphi$ ; data
2:  $url\_groups$  = Collect URL groups by aligning HTML
   DOM tree of  $sp$ ;
3: foreach  $ug$  in  $url\_groups$  do
4:    $ug.anchor\_len$  = Total anchor text length in  $ug$ ;
5: end foreach
6:  $it\_group$  = arg max(  $ug.anchor\_len$  ) in  $url\_groups$ ;
7:  $it\_group.DstPageType$  = Majority page type of the des-
   tination pages of URLs in  $ug$ ;
8: if  $it\_group.DstPageType$  is  $INDEX\_PAGE$ 
9:    $it\_group.UrlType$  =  $INDEX\_URL$ ;
10: else if  $it\_group.DstPageType$  is  $THREAD\_PAGE$ 
11:    $it\_group.UrlType$  =  $THREAD\_URL$ ;
12: else
13:    $it\_group$  =  $\varphi$ ;
14: end if
15: return  $it\_group$ ;
```

4.2.2. 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 is called single page-flipping URL. Such URLs cannot be detected using the “connectivity” metric. To address this shortcoming, it 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. It uses 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.

Algorithm PageFlippingUrlDetection

Input: sp : an index page or thread page

Output: pf_group : a group of page-flipping URLs

```
1: let  $pf\_group$  be  $\varphi$ ;  
2:  $url\_groups$  = Collect URL groups by aligning HTML  
   DOM tree of  $sp$ ;  
3: foreach  $ug$  in  $url\_groups$  do  
4:   if the anchor texts of  $ug$  are digit strings  
5:      $pages$  = Download( URLs in  $ug$  );  
6:     if  $pages$  have the similar layout to  $sp$  and  $ug$   
       appears at same location of  $pages$  as in  $sp$   
7:        $pf\_group$  =  $ug$ ;  
8:       break;  
9:     end if  
10:  end if  
11: end foreach  
12: if  $pf\_group$  is  $\varphi$   
13:  foreach  $url$  in outgoing URLs in  $sp$   
14:     $p$  = Download(  $url$  );  
15:     $pf\_url$  = Extract URL in  $p$  at the same location as  
        $url$  in  $sp$ ;  
16:    if  $pf\_url$  exists and  $pf\_url.anchor == url.anchor$   
       and  $pf\_url.UrlString != url.UrlString$   
17:      Add  $url$  and  $cand\_url$  into  $pf\_group$ ;  
18:      break;  
19:    end if  
20:  end foreach  
21: end if  
22:  $pf\_group.UrlType$  =  $PAGE\_FLIPPING\_URL$ ;  
23: return  $pf\_group$ ;
```

4.3 SPARSE SOCIAL DIMENSIONS

In this section, to first show one toy example to illustrate the intuition of communities in an “edge” view and then present potential solutions to extract sparse social dimensions.

4.3.1 COMMUNITIES IN AN EDGE-CENTRIC VIEW

4.3.2 EDGE PARTITION VIA LINE GRAPH PARTITION

4.3.3 EDGE PARTITION VIA CLUSTERING EDGE INSTANCES

4.3.1 COMMUNITIES IN AN EDGE-CENTRIC VIEW

Though SocioDim with soft clustering for social dimension extraction demonstrated promising results, its scalability is limited. A network may be sparse i.e., the density of connectivity is very low), whereas the extracted social dimensions are not sparse. Let’s look at the toy network with two communities in Figure 1. Its social dimensions following modularity maximization are shown in Table 2. Clearly, none of the entries is zero.

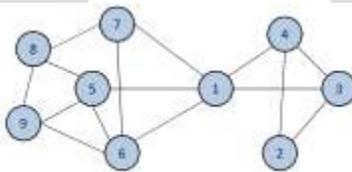


Figure. No.1: 1 Toy example

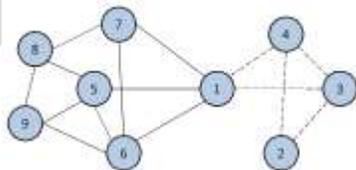


Figure. No: 2 Edge cluster

Then a network expands into millions of actors, a reasonably large number of social dimensions need to be extracted. The corresponding memory requirement hinders both the extraction of social dimensions and the subsequent discriminative learning. Hence, it is imperative to develop some other approach so that the extracted social dimensions are sparse.

5. SYSTEM DESIGN

5.1 Module Design

The thesis contains the following modules.

The following modules are present in the thesis

1. Index Url And Thread Url Training Sets
2. Page-Flipping Url Training Set
3. Entry Url Discovery
4. Create Graph
5. Convert To Line Graph
6. Algorithm Of Scalable K-Means Variant
7. Algorithm For Learning Of Collective Behavior
8. Sentiment Analysis
 - 1) Forum Topic Download
 - 2) Parse Forum Topic Text And Urls
 - 3) Forum Sub Topic Download
 - 4) Parse Forum Sub Topic Text And Urls

1. Index Url And Thread Url Training Sets

The homepage of a forum which contains a list of boards and is also the lowest common ancestor of all threads. A page of a board in a forum, which usually contains a table-like structure; each row in it contains information of a board or a thread. 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. The only way to distinguish index URLs from thread URLs is the type of their destination pages. Therefore, user needs a method to decide the page type of a destination page.

2. 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 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 more than one page-flipping URL in one page.

3. Entry Url Discovery

An entry URL needs to be specified to start the crawling process. To the best of our knowledge, all previous methods assumed that a forum entry URL is given. In practice, especially in web-scale crawling, manual forum entry URL annotation is not practical. Forum entry URL discovery is not a trivial task since entry URLs vary from forums to forums.

4. Create Graph

In this module, nodes are created flexibly. The name of the node is coined automatically. The name should be unique. The link can be created by selecting starting and ending node; a node is linked with a direction. The link name given cannot be repeated. The constructed graph is stored in database. Previous constructed graph can be retrieved when ever from the database.

5. Convert To Line Graph

In this module, from the previous module's graph data, line graph is created. The edge details are gathered and constructed as nodes. The nodes with same id in them are connected as edges.

6. Algorithm Of Scalable K-Means Variant

In this module, the data instances are given as input along with number of clusters, and clusters are retrieved as output. First it is required to construct a mapping from features to instances. Then cluster centroids are initialized. Then maximum similarity is given and looping is worked out. When the change is objective value falls above the 'Epsilon' value then the loop is terminated.

7. Algorithm For Learning Of Collective Behavior

In This Module, The Input Is Network Data, Labels Of Some Nodes And Number Of Social Dimensions; Output Is Labels Of Unlabeled Nodes.

8. Sentiment Analysis

1) Forum Topic Download

In This Module, The Source Web Page Is Keyed In (Default: [Http://Www.Forums.Digitalpoint.Com](http://www.forums.digitalpoint.com)) And The Content Is Being Downloaded. The HTML Content Is Displayed In A Rich Text Box Control.

2) Parse Forum Topic Text And Urls

In This Module, The Downloaded Source Page Web Content Is Parsed And Checked For Forum Links. The Links Are Extracted And Displayed In A List Box Control. Also The Link Text Are Extracted And Displayed In Another List Box Control.

3) Forum Sub Topic Download

In this module, all the forum links pages in the source web page are downloaded. The HTML content is displayed in a rich text box control during each page download.

4) Parse Forum Sub Topic Text And Urls

In this module, the downloaded forum pages web content are parsed and checked for sub forum links. The links are extracted and displayed in a list box control. Also the link text are extracted and displayed in another list box control.

6. RESULT AND DISCUSSION

ANALYZING AVERAGE POST PER FORUM AND AVERAGE SENTIMENTAL VALUE

Forum Id	Forum Title	Threads count	Post Count	Average Post Per forum	Average sentiment value per forum
1	Google	4	1340	335	0
34	Google+	51	1158	22	1
37	Digital Point Ads	50	708	14	1
38	Google AdWords	53	684	12	0
39	Yahoo Search Marketing	50	1240	24	1
44	Google	50	2094	41	0
46	Azoogle	51	1516	29	0
49	ClickBank	50	1352	27	0
52	General Business	51	1206	23	0
54	Payment Processing	52	1782	34	0
59	Copywriting	51	526	10	0
62	Sites	53	504	9	1
63	Domains	51	78	1	1
66	eBooks	51	484	9	1
70	Content Creation	50	206	4	1

71	Design	50	498	9	1
72	Programming	51	202	3	1
77	Template Sponsorship	47	94	2	1
82	Adult	51	30	0	1
83	Design & Development	6	0	0	1
84	HTML & Website Design	52	254	4	1
85	CSS	50	110	2	1
86	Graphics & Multimedia	54	79	1	0

Table No: 5.3 Analyzing Average Post Per Forum And Average Sentimental Value

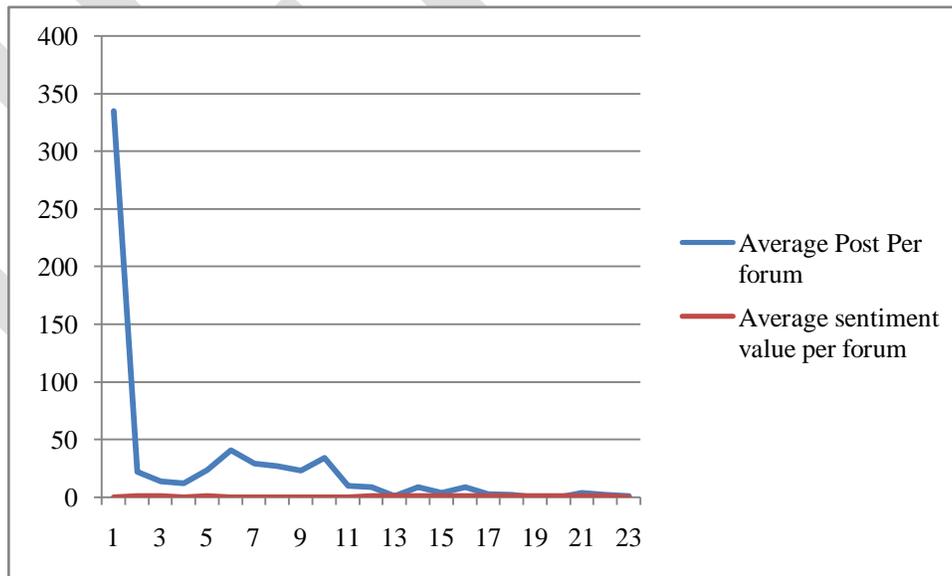


CHART NO: 5.3 CHART REPRESENTATION FOR ANALYZING AVERAGE POST PER FORUM AND AVERAGE SENTIMENTAL VALUE

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.

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this thesis 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.

6.2.SCOPE FOR FUTURE ENHANCEMENTS

In the future, 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. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. L. Zhang, B. Liu, S.H. Lim, and E. O'Brien-Strain, "Extracting and Ranking Product Features in Opinion Documents," Proc. 23rd Int'l Conf. Computational Linguistics, pp. 1462-1470, 2010.

7. 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.
8. 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.
9. J.-M. Yang, R. Cai, Y. Wang, J. Zhu, L. Zhang, and W.-Y. Ma, "Incorporating Site-Level Knowledge to Extract Structured Data from Web Forums," Proc. 18th Int'l Conf. World Wide Web, pp. 181-190, 2009.
10. 28] Y. Zhai and B. Liu, "Structured Data Extraction from the Web based on Partial Tree Alignment," IEEE Trans. Knowledge Data Eng., vol. 18, no. 12, pp. 1614-1628, Dec. 2006.
11. [29] J. Zhang, M.S. Ackerman, and L. Adamic, "Expertise Networks in Online Communities: Structure and Algorithms," Proc. 16th Int'l Conf. World Wide Web, pp. 221-230, 2007.
12. Blog, <http://en.wikipedia.org/wiki/Blog>, 2012.
13. "ForumMatrix," <http://www.forummatrix.org/index.php>, 2012.
14. Hot Scripts, <http://www.hotscripts.com/index.php>, 2012.
15. Internet Forum, http://en.wikipedia.org/wiki/Internet_forum, 2012.
16. "Message Boards Statistics," <http://www.big-boards.com/statistics/>, 2012