WEB FORUM CRAWLER FOR USER SENTIMENTAL ANALYSIS

[1]Anju A, [2]Dr. Saravanan T. M
[1]Final Year PG Scholar,[2]Assistant Professor
[1]Master of Computer Applications,
[1]Kongu Engineering College,Perundurai,Tamilnadu,India

ABSTRACT

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. 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 project aims to learn to predict collective behavior in social media. In addition, the project 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. As a result, the behavior of individuals is collected through their posts in a forum and then they are classified as positive/negative posts. The cases are encoded in terms of features in some numerical form, requiring a transformation from text to numbers and assign the positive and negative values to each word to classify the word in the document.

Keywords : **Data mining, Sentiment Analysis, K Means clustering**

1. INTRODUCTION

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.

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 contentor 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.

Crawling Policy

The behavior of a Web crawler is the outcome of a combination of policies:

- A selection policy that states which pages to download,
- A re-visit policy that states when to check for changes to the pages,
- A politeness policy that states how to avoid overloading Web sites, and
- A parallelization policy that states how to coordinate distributed web crawlers.

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.

II. RELATED WORK

SERGEY BRIN and LAWRENCE PAGE present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms

RUI CAI, JIANG-MING YANG and WEI LAI, YIDA WANG present Web forum crawling using study the problem, which is a very fundamental step in many Web applications, such as search engine and Web data mining. As a typical user-created content (UCC), Web forum has become an important resource on the Web due to its rich information contributed by millions of Internet users every day. However, Web forum crawling is not a trivial problem due to the in-depth link structures, the large amount of duplicate pages, as well as many invalid pages caused by login failure issues. In this paper, we propose and build a prototype of an intelligent forum crawler, iRobot, which has intelligence to understand the content and the structure of a forum site, and then decide how to choose traversal paths among different kinds of pages

ANIRBAN DASGUPTA and RAVI KUMAR Anirban Dasgupta Ravi Kumar Amit Sasturka stated that a large fraction of the URLs on the web contain duplicate (or near-duplicate) content. De-duping URLs is an extremely important problem for search engines, since all the principal functions of a search engine, including crawling, indexing, ranking, and presentation, are adversely impacted by the presence of duplicate URLs. Traditionally, the de-duping problem has been addressed by fetching and examining the content of the URL; their approach here is different. Given a set of URLs partitioned into equivalence classes based on the content (URLs in the same equivalence class have similar content), we address the problem of mining this set and learning URL rewrite rules that transform all URLs of an equivalence class to the same canonical form

MONIKA HENZINGER describe duplicate and near-duplicate web pages are creating large problems for web search engines: They increase the space needed to store the index, either slow down or increase the cost of serving results, and annoy the users. Thus, algorithms for detecting these pages are needed. A naive solution is to compare all pairs to documents. Since this is prohibitively expensive on large datasets, proposed of comparisons. Both algorithms work on sequences of adjacent characters. In this paper started to use word sequences to detect copyright violations. In this research and focused on scaling it up to multi-gigabyte databases are used. Word sequences to efficiently and near-duplicate web pages.

They are proposed system performed an evaluation of two near-duplicate algorithms on 1.6B web pages. Neither performed well on pages from the same site, but a combined algorithm did without sacrificing much recall. Two changes might improve the performance of Alg. B and deserve further study: (1) A weighting of shingles by frequency and (2) using a different number k of tokens in a shingle. For example, following one could try k = 5. However, recall that 28% of the incorrect pairs are caused by pairs of pages in two databases on the web. In these pairs the difference is formed by one consecutive sequence of tokens. Thus, reducing k would actually increase the chances that pairs of pages in these databases are incorrectly identified as near-duplicates.

HEMA SWETHA KOPPULA and KRISHNA P. LEELA stated that Presence of duplicate documents in the World Wide Web adversely affects crawling, indexing and relevance, which are the core building blocks of web search. In this paper, they present a set of techniques to mine rules from URLs and utilize these rules for de-duplication using just URL strings without fetching the content explicitly. Their technique is composed of mining the crawl logs and utilizing clusters of similar pages to extract transformation rules, which are used to normalize URLs belonging to each cluster. Preserving each mined rule for de-duplication is not efficient due to the large number of such rules. They present a machine learning technique to generalize the set of rules, which reduces the resource footprint to be usable at web-scale. The rule extraction techniques are robust against web-site specific URL conventions.

L. ZHANG, B. LIU and S.H. LI stated that An important task of opinion mining is to extract people's opinions on features of an entity. For example, the sentence, "The GPS function of Motorola Droid" expresses a positive opinion on the "GPS function" of the Motorola phone. "GPS function" is the feature. The paper focuses on mining features. Double propagation is a state-of-the-art technique for solving the problem. It works well for medium-size corpora. However, for large and small corpora, it can result in low precision and low re-call. To deal with these two problems, two improvements based on part-whole and "no" patterns are introduced to increase the recall. Then feature ranking is applied to the extracted feature candidates to improve the precision of the top-ranked candidates.

III. SYSTEM METHODOLOGY

3.1 TERMINOLOGY

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

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.

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.

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).

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.

3.2. 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.

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; we 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, we need a method to decide the page type of a destination 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 φ;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 = φ;
14: end if
15: return it_group;
```

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.

3.2.2. 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 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:

- Their anchor text is either a sequence of digits such as 1, 2, 3, or special text such as "last."
- They appear at the same location on the DOM tree of their source page and the DOM trees of their destination pages.
- 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.
- 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 φ;
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 φ
13:   foreach url in outging 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;
```

IV PROPOSED SYSTEM METHODOLOGY

4.1. INDEX URL AND THREAD URL TRAINING SETS

The homepage of a forum which is 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.

4.2. 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.

4.3. 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.

4.4. 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.

4.5. 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.

4. 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.

4.7. 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.

The following steps are worked out.

- Convert network into edge-centric view.
- Edge clustering is performed.
- Construct social dimensions based on edge partition. A node belongs to one community as long as any of its neighboring edges is in that community.
- Apply regularization to social dimensions.
- Construct classifier based on social dimensions of labeled nodes.
- Use the classifier to predict labels of unlabeled ones based on their social dimensions.

4.8. SENTIMENT ANALYSIS

1) FORUM TOPIC DOWNLOAD

In this module, the source web page is keyed in (default: 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.

V. EXPERIMENTAL RESULT

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.

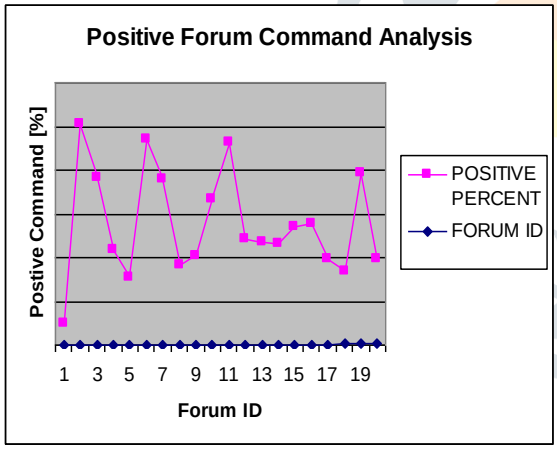
Table 6.1 Positive Forum Command Analysis (Count)

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	45	1904

The following **Fig 6.1** 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 6.1 Positive Forum Command Analysis



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 6.2** 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 6.1 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 6.2** 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.

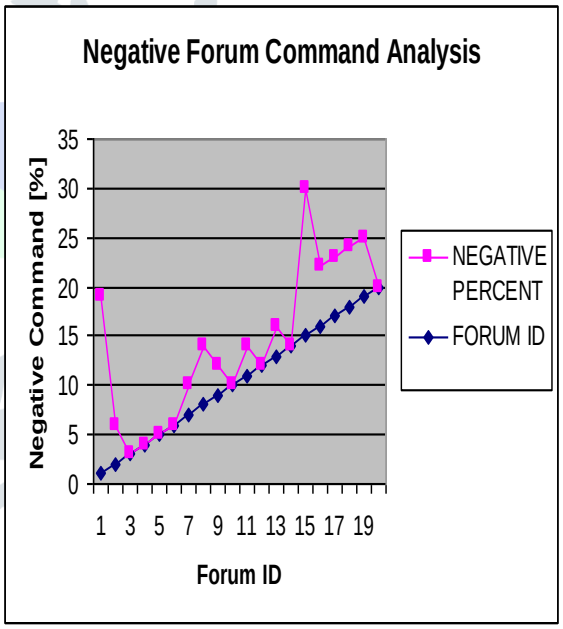


Table 6.2 Negative Forum Command Analysis (Count)

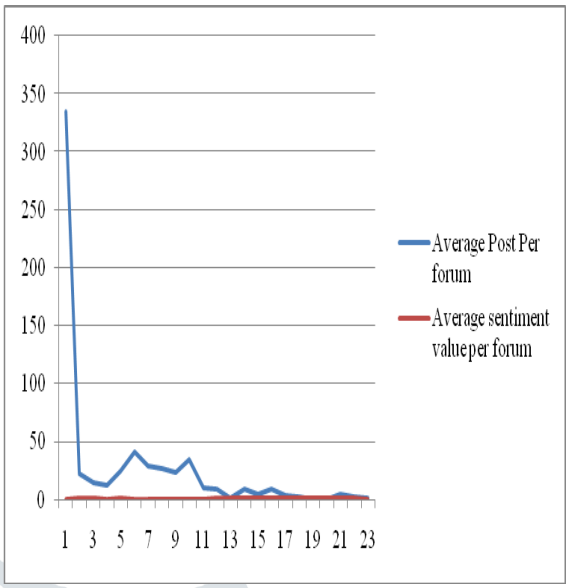
ANALYZING AVERAGE POST PER FORUM AND AVERAGE SENTIMENTAL VALUE

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

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.



VI. CONCLUSION

In this proposed method 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.

VI SCOPE FOR FUTURE ENHANCEMENTS

In 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. 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. 29thAnn. 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.
17. nofollow, http://en.wikipedia.org/wiki/Nofollow, 2012. [7] "RFC 1738—Uniform Resource Locators (URL)," http://www.ietf.org/rfc/rfc1738.txt, 2012.
18. Session ID, http://en.wikipedia.org/wiki/Session_ID, 2012.
19. "The Sitemap Protocol," http://sitemaps.org/protocol.php, 2012.
20. "The Web Robots Pages," http://www.robotstxt.org/, 2012.