IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 6, June 2016

An Efficient Multi-Web Search Engines Results based on probability Cluster

G. Sharmila[1], Mr. M. Siva Kumar M.E[2]

M.E Computer Science and Engineering, SVS College of Engineering, Coimbatore, Tamilnadu, India[1]

Assistant Professor, SVS College of Engineering, Coimbatore, Tamilnadu, India[2]

Abstract: Web search is considered the mainly valuable place for information retrieval and knowledge discovery. Web is ever more used not only to find answers to specific information needs but also to carry out various tasks, attractive the capability of current web search engines with effective and resourceful techniques for web service retrieval and selection becomes an important issue. Existing web search result based on keyword matching in single search engine only it will not give accurate result of the query. The Proposed system efficient multiple web search results based on probability clustering system that enhances search results performance (i) multi search engine method is lists of web results returned by user queries to search engines. Ii) Probability k means cluster using search results term based cluster based on this approach, in this system, a mechanism is being proposed that provides ordered results in the form of likelihood based clusters in agreement with users query. An efficient cluster method is also proposed that orders the results according to both the relevancy and the importance of web results. Web search result clustering has been emerged as a method which overcomes this problem of conventional information retrieval (IR) machine. It is the probability clustering of results returned by the search engines into meaningful, thematic groups. This paper gives a succinct overview and categorizes various techniques that have been used in clustering of web search results.

Keywords: web mining; Information Retrieval (IR); Clustering; Ranking; Text mining; web search engine.

I. INTRODUCTION

Effective web searchers employ a variety of techniques to find the material they're looking for. They build on background knowledge to form their query, think about what good results might be, try multiple keywords and read through results carefully. Moreover the process of retrieval is extremely affected by the unclear query put up by the standard user. Although today's search engines are smarter than earlier, unclear queries are still a main problem. To answer all the possible meaning of an unclear query, search engines return too many results which are not necessarily related to the user's need. Usually user has to cross several search result pages to get to the preferred result. A way of support users in finding what they are looking for quickly is to group the search results by topic value. The user does not have to reformulate the query, but can just click on the topic most accurately relating his or her specific information need. This group of result is called cluster. More specifically, it is a process of grouping similar web result documents into clusters so that web documents of one cluster are different from the web documents of other clusters. There are many exiting web clustering engines available on the web (Carrot2, Vivisimo, SnakeT, Grouper etc) which give the search results in form of clusters the search result. A web clustering engine takes the result, returned by the search engine as input and performs clustering and classification on that result. This process is usually seen as balancing rather than alternative and different to the search engine [1]. The main goal for web search result clustering is not to improve the actual ranking, but to give the user a quick overview of the results. Having divided the result set into clusters, the user can quickly narrow down his search further by selecting a cluster. This resembles query modification, but avoids the need to query the search engine for each step. Web search result clustering has been the focus of IR population since the appearance of web search engine. Therefore several works has been done in this area. The Scatter/Gather system by [2] is held as the predecessor and theoretical father of all web search result clustering engine. Web Search engine is the most normally used tool for information retrieval on the web; however, its current status is far from satisfaction for several possible reasons [3], such as different users have different requirements and potential for search results; sometimes queries cannot be expressed clearly just in several keywords; Synonymous and polysemous words make searching more complicated etc.

The rest of the paper is organized as follows. Section 2, gives an overview of related work. Section 3 presents proposed approach. In Section 4, deal with some performance to validate proposed approach. Conclusion is presented in Section 5.

II. RELATED WORKS

The Existing system was the person who coined the term web mining primary time [1]. At first two different approaches were taken for important web mining results. primary was a "process-centric view", which defined web mining as a series of different processes as resource discovery, information retrieval and simplification [1,10]whereas, second was a "data centric view", which defined web mining in terms of the type of data that was being used in the web result mining process [2].

The existing system definition has become more satisfactory, as is clear from the approach adopted in most research papers [6]. Web mining is also a irritated point of database, information retrieval and artificial intelligence. The most common way of in place of web documents is using the Vector Space Model (VSM) algorithm to classify the document [12], where each document is represented as a feature vector, which length correspond to the number of unique attributes used for on behalf of web documents in the collection. In VSM vector component, that is, each feature has a linked weight which indicates the significance of that attribute to characterize or represent the web document. Web mining can be categorized into three different classes based on which part of the web is to be mined. Web content mining, Web structure mining and Web usage mining [4, 7, and 9]. Oren Zamir and Oren Etzioni [1] in their research listed the key supplies of web document clustering methods as relevance, brow able summary, overlap, snippet tolerance,web result speed and accuracy. They have given STC (Suffix Tree Clustering) algorithm which creates clusters based on expression shared between web documents. Fresno and Ribeiro in 2004 presented an Analytical Combination of Criteria (ACC) algorithm to represent web pages mining. It is based on a linear mixture of different heuristic criteria within the VSM. The criteria used by ACC word frequency in the title of the document, Emphasis: web content word frequency in highlighted text segments, Position: word positions in a web document & Frequency: word frequency in the web document. Fresno in 2006 proposed an alternative way of combining them in a non-linear way. In this case, a fuzzy logic based system is employed to define the expert knowledge about how to combine these web search techniques. In [13] paper author present a method for extracting news content from the Web search engine , based on the visual awareness of human users and try to simulate how human beings know the information found in web news by using a function based object model. The objects of this model can be of different type first one information object, navigation object, communication object and beautification object. Researchers at first proposed web document clustering for information retrieval and web search to improve search performance by validate the cluster theory, which states that web documents in the same cluster behave similarly with respect to significance to information needs. In recent years, researchers have used clustering to organize search results, creating a cluster based web search interface as an alternative appearance to the list interface. Web document clustering is widely applicable in areas such as search engines, web mining and information retrieval. Most web document clustering methods perform several pre-processing steps including stop words removal and stemming on the web document set [3].

Most of the existing web document clustering algorithm worked on BOW (Bag of Words) model [7].Each web document is represented by a vector of frequencies (TF) of residual terms within the web document. Some web document clustering algorithms utilize an extra pre-processing step that divide the actual term frequency by the overall frequency of the term in the entire web document set(TF-IDF).It has great potentials in application like object recognition, image segmentation and information filtering and retrieval [4]. Most of the clustering techniques go down into two major categories, and these are the hierarchical clustering and the partitioned clustering used in exiting researcher [4].

The existing author Scatter/gather describes in [11] was an early cluster based web document browsing method that performs post retrieval clustering on top-ranked web documents returned from a traditional information retrieval system.

A. Limitations of Web Search

In existing system have several limitations with a huge growth of the Internet it has become very difficult for the users to find relevant web result. In reply to the user's query, currently available search engines return a without ranked list of web documents along with their incomplete content. If the query is universal, it is very difficult to identify the specific web matched document which the user is involved in. The users are forced to filter through a long list of off-web-topic documents. Moreover, inside relationships among the web documents in the search result are hardly ever presented and are left for the user. In existing standard information retrieval systems rely on two orthogonal paradigms: the textual based similarity with the query on one give and a query independent measure of each web page's importance on the other hand. Though, these systems generally lack user model and thus are far from being most favorable i.e. Different users may submit exactly the same query even though they have different intention. The most famous examples of such unclear queries include bass, java (programming language, island or coffee), jaguar (animal, car or Apple software) and IR application (Infrared application or Information Retrieval application).

III. PROPOSED APPROACH

The Proposed system clustering of web search results based on probability k means cluster has been implemented in the area of web information retrieval (IR). The objective of clustering search result is to give user and thought of what the result contains. This idea is in the form of clusters. Clustering in context of web search result means organizing query result pages into groups based on their similarity between each other. Search result clustering techniques exact to the search engine result can be generally classified as content based and topology based clustering. Web document clustering can be classified as the content-based clustering. Graph based clustering can be categorized as topology-based clustering. The contribution of the proposed web search engine has four components:

- Web document processor indexes new documents. Indices are a mapping between words and what documents they come into view in. Most web engines are spider-based, so a crawl of the web for new documents and the updating of the index are automated.

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 6, June 2016

- Query processor inspects a user's query and translates it into impressive internally important.
- Web result matching function uses the above internally important representation to extract web documents from the index and finally form the cluster.
- Web ranking method positions the more-relevant web documents on top, using click based relevance measure.

A. Design Multiple Search engine

The majority known common search engines are Google and Yahoo!, but one of the oldest search engines is AltaVista and many more search engine are used in this proposed system. The existing search engines have weaknesses; even Google search use keyword based matching the user query. This part represents a real reason for construction more search engine. A scalable distributed warehouse is used to store the crawled collection of Web pages. Strategy for physical group of pages on the storage devices, distribution of pages across machines, and mechanisms to integrate newly crawled pages, is important issues in the design of this web repository. The repository supports both random and stream-based access modes. Random access allows individual pages to be retrieved based on an internal page identifier. Stream-based access allows all or a significant subset of pages to be retrieved as a stream. Query-based access to the pages and the computed features is provided via the web Base query engine. Different the traditional keyword based queries supported by existing search engines, queries to the web base query engine can occupy predicates on both the content and link structure of the web documents. In selection of search engines ten search engines were selected to conduct our experiment. They are All the Web, AltaVista, google, yahoo, clusty, you tube, file tube, citeceer etc., to name a few. At first, the search engines were selected and the user query is submitted to all search engines under thought. The queries covered a broad range of topics. The topics are as follows: Computer science, education, Internet, literature, music, plants, sports, travel etc. A single query performs each ten search engine and crawler the web result. The precision of content of these pages is compared to give the result.

B. Web Crawler

Web crawler development continues until all reachable content has been gathered, until the refresh interval is complete or until another configuration parameter limiting the scope of the crawl is reached. There are many different ways to alter the design to suit a specific web crawling scenario.

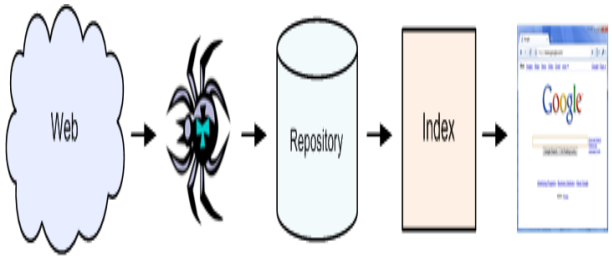

Fig.1 Web is crawled

a. Controller Module: This module focuses on the Graphical User Interface (GUI) designed for the web crawler and search result is accountable for scheming the operations of the web crawler. The GUI enables the user to enter the start query with URL, enter the maximum number of result you want, view the URL's that are being fetch. It controls the fetcher and parser.

b: Fetcher Module - This module process by fetching the page according to the query start URL specified by the user. The fetcher module also retrieves all the links in a particular page based on user query and continues doing that until the maximum number of total no of results is reached.

c. Parser Module – The last module parses the URL's fetched by the Fetcher module and saves the matched contents of those pages to the GUI. After that indexer create index in the database to organize the data by categorize them. The indexer extracts all the information from each and every web document and stores it in a database. All high quality search engines index each and every word in the web documents and give a unique word search result Id. Then the word occurrences, which efficient search engines call "hits," are checked, recording all the words, including their post in the web document, their font size and capitalization.

C. Web Result Filtering

The proposed techniques used to filter the search result based on Bloom Filter. A Bloom filter of a set U is implemented as an array of m bits. Each element u ($u \in U$) of the set is hash using used defined k independent hash function h1. . . hk. Each hash function hi(u) for $1 \leq i \leq k$ maps to one bit in the array {1 . . .m}. Thus, when an element is added to the set, it sets k bits, each bit corresponding to a hash function, in the Bloom filter array to 1. If a bit was already set it stays 1. For set relationship checks, Bloom filters may yield a false positive, where it may become visible that an element v is in U even though it is not. From the investigation in, given n = |U| and the Bloom filter size m, the optimal value of k that minimize the false positive likelihood, pk, where p denotes that likelihood that a given bit is set in the Bloom filter, is k = m n ln 2. Previously, Bloom filters have primarily been used for finding set-membership.

Here finding similar web result documents, the In case the two search result share a large number of 1's (bit-wise AND) they are noticeable as similar. In this case, the bit-wise AND can also be apparent as the dot creation of the two bit vectors. If the set bits in the Bloom filter of a web document are a complete subset of that of another filter then it is highly probable that the web document is included in the other. Web pages are characteristically composed of remains, either static ones, or dynamic. When target pages for a similarity based "clustering", the test for similarity should be on the fragment of interest and not the entire page.

Bloom filters, when applied to similarity discovery, have several compensation. First, the density of Bloom filters is very attractive for storage and transmission whenever we want to minimize the meta-data expenses. Second, Bloom

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 6, June 2016

filters enable fast comparison as matching is a bitwise-AND operation. Third, since Bloom filters are a complete representation of a set rather than a deterministic sample they can decide inclusion efficiently.

D. Cluster Web Result

Probability K-Means clustering is one of the most common and efficient clustering algorithms. It clusters each web topic into one of K groups. K is a pre-determined positive integer that can be obtained by arbitrary selection or by some other topic model processes that observe the data relationships iteratively. Once the number of final clusters is decided, it needs to pick up K data points from data web collection as the initial centroids for the first task of data web topic. The assignment of all data web topic to different clusters is performed iteratively awaiting some stop condition is reached. The main principle of probability K-Means is described as follows:

1. Pre-determine the user defined K values of final clusters and randomly select the K web content as initial cluster centroids.
2. Allocate each web content topic to the cluster that is closest to.
3. Re-compute K centroids after all web data retrieved have been assigned to corresponding clusters.
4. Repeat the step 2 and 3 until the k means stop condition is reached, e.g. the certain amount of iteration is finished or all cluster centroids don't change any more between iterations and etc.

Cluster Distance measure is usually the most common similarity metrics probability K-Means clustering uses, such as Squared Euclidean distance measure as shown in the Equation 1, where x1, x2, ..., xn is the representation of point X and y1, y2, ..., yn is the representation of point Y. But both Euclidean distance and Euclidean distance don't consider the normalization, therefore, K-Means clustering uses cosine similarity metrics that is described previously in the section of "Vector Space Model".

Equation 1

$$d = \sum_{i=1}^{n} (x_i - y_i)^2$$

Clustering system usually consists of web documents crawling, ranking and clustering as its essential procedures. Our probability K-Means clustering method is implemented on top of Apache Lucene indexing, ranking creation and probability K-Means clustering components.

E. Probabilistic Cluster

Here assume the pages are ordered by the search engine in order of their indices: 1,2,3,... N. Then the following two values represent the expected likelihood that users finally click-through and the expected number of pages views per user until a click-through.

E[probability of success] = p1 + (1 ¡ p1)[p2 + (1 ¡ p2)[p3 + ….. E[search time] = 1 + (1 ¡ p1)[1 + (1 ¡ p2)[1 + …N

It is reasonable to assume that maximizing the first of these values and minimizing the next are equally main objectives for a web search engine. Now, clearly with respect to our model and supposition the above equation, the probability of a click-through will be the same at every time. However, if we relax supposition, this quantity does not remain stable as we reorder web pages. Likewise, the expected number of page views changes as reorder web pages, regardless of supposition. In any model, if we can simply recognize the pi values then we can optimize with respect to both of these objectives by simply ordering the pi values in decreasing order. The probability of a click-through in m steps can be rewritten E[probability of success] = 1 ¡ (1 ¡ p1)(1 ¡ p2)……(1 ¡ pm) This value is falling in pi, so we want the highest pi's included for all sets of m steps. Thus, we want pi ranked in decreasing order to make the most of this probability for all m. Expected number of page examinations can also be rewritten as E[search time] = 1 + (1 ¡ p1) + (1 ¡ p1)(1 ¡ p2) +…. + (1 ¡ p1)….(1 ¡ pk) so for any ordering of the pages, if you swap pages i and j where i was originally placed before j, the only terms in this sum that change are those that include a pi term and no pj term. These terms all decrease if pj > pi and increase if pj < pi. Thus, to minimize expected number of page examinations, we must order by largest pi.

We begin by modeling the system. here assume:

1. There are k web pages.
2. The search engine distinguish between pages by topic, thus every user query is equally relevant to all pages.
3. Each page i have an intrinsic value parameter pi which represents the probability that any given user, upon examining page i's list on the search engine, will click-through to page i.
4. The search engine produces for each user a planned list of pages. Users examine these pages in order until they make a decision to click-through to a page. Once the user clicks-through to that page, the user is done.
5. Users will continue examining pages until they have either clicked-through to a page, or discarded all pages.

IV. EXPERIMENTAL RESULTS

Here multiple web search engine. These strategies are ranked with cluster and click based ranking algorithm as well as with a click ranking approach. The proposed algorithm mainly deals with the concept of when the submitted query give the predictable result then the links returned by the given query gives out the best result with clustering. Experimental results showed a better result by using this proposed algorithm against click based ranking. The performance measure can be applied by using Precision and recall method as follows.

V. PERFORMANCE MEASURE

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Where,

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 6, June 2016

- True positives (TP) - number of reviews correctly label as belong to exacting class (positive/negative).
- False positives (FP) - number of result incorrectly cluster as belonging to particular query.
- False negatives (FN) - number of cluster were not label as belonging to the particular query but should have been labeled.

Table 5.1: Number of user Query Vs Precision

Algorithms	**5**	**10**	**15**	**20**	**25**
Existing	**0.38**	**0.28**	**0.21**	**0.18**	**0.15**
Proposed System	**0.49**	**0.41**	**0.37**	**0.27**	**0.19**

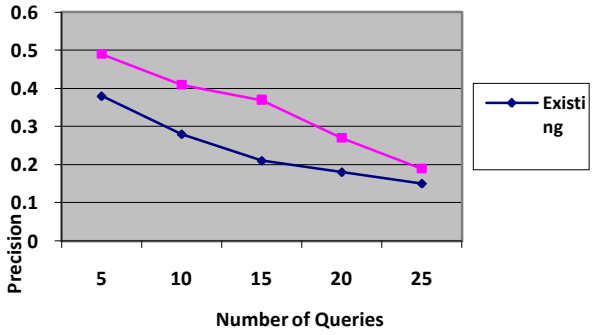

Figure 5.1 Number of Query Vs Precision

Table 5.2: Recall Vs Precision

Algorithms	0.1	0.2	0.3	0.4	0.6	0.7	0.8	0.9	1
Existing	0.81	0.74	0.65	0.6	0.53	0.47	0.3	0.21	0.1
Proposed	0.95	0.81	0.74	0.68	0.59	0.51	0.43	0.33	0.2

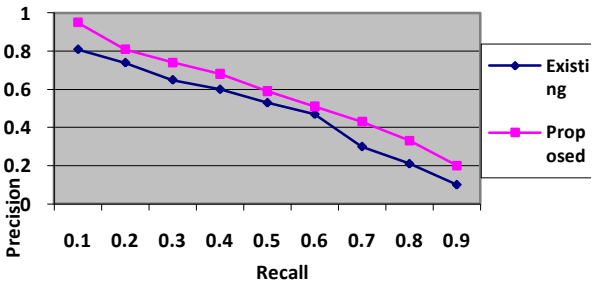

Figure 5.2 Comparison of Averaged Precision-Recall (PR)

Table 5.3: The relevancy values for the query "java" produced by PageRank and clustering value

Cluster Value	Existing		Proposed	
	PageRank	Cluser	PageRank	Cluser
5	2	5	4	15
10	3	8	5	24
20	5	12	8	32
30	8	14	12	46
40	10	15	14	54
80	15	18	25	62
100	18	21	31	79

Table 5.4: Different Search engine with Result

Web Search Engine URL	User Query	Normal Query Result	Cluster Based web result
Yahoo	Java	50	80
wikipedia	Java	30	60
Isohunt	Java	10	40
torrenz	java	70	80

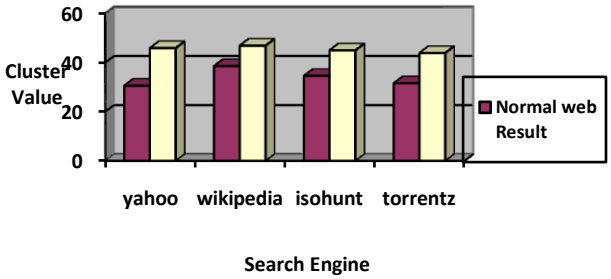

Figure 5.4 Different search engine with Cluster

Table 5.5: Different Query with Cluster Size

Topic	Number of final clusters	Singleton Cluster	Maximum Cluster Size	Number of clusters with size >3
Data mining	15	66	13	5
Data mining	14	60	20	4
Data mining	12	53	20	8
Java	21	129	89	6
Java	23	105	107	8

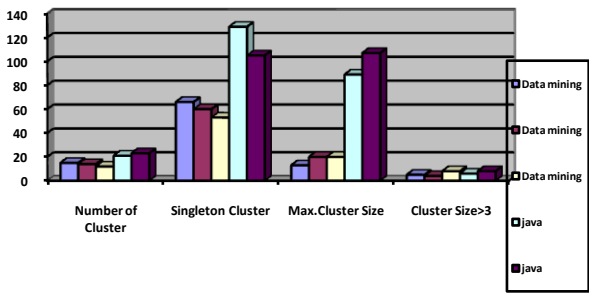

Figure 5.5 Different search engine with query

VI. CONCLUSION

Given the important role of search engines in the World Wide Web, here improve the crawling process employed by multiple search engines with the goal of improving the quality of the service they provide to clients. Our analysis of the cluster the web result and ranking as done, and the metric of embarrassment, which we introduced as a preferable goal. The next-generation Web architecture represented by the Semantic Web will provide adequate instruments for improving search strategies and enhance the probability of seeing the user query satisfied without requiring tiresome manual refinement. Future enhancement of Particle Swarm Optimization method based upon the concept of Swarm Intelligence is being implemented in high-dimensional sequence clustering analysis for web usage mining.

REFERENCES

[1] Delay Tolerant Networking Research Group. http://www.dtnrg.org.
[2] Conti, M., Crowcroft, J., Giordano, S., Hui, P., Nguyen, H.A., & Passarella, A.(2008). Minema. Hugo Miranda, Luis Rodrigues,Benoit Garbinato (Ed.), "Routing issues in Opportunistic Networks". Springer.
[3] Mamoun H. M., "Efficient Routing Scheme for Opportunistic Networks", International Journal of Engineering and Technology, Vol. 2, No 6, pp. 940-945, June 2012.
[4] Hemal Shah, Yogeshwar P. Kosta, "Exploiting Wireless Networks, through creation of Opportunity Network – Wireless-Mobile-Adhoc-Network (W-MAN) Scheme", International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Volume.2, No.1, March 2011,99-110.
[5] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks", Tech. Rep. CS-2000-06, CS Dept., Duke University, April 2000.
[6] A. Lindgren et al, "Probabilistic Routing in Intermittently Connected Networks", Mobile Comp. and Comm. Rev, vol. 7, no. 3, pp. 19- 20, July 2003.
[7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: Efficient routing in intermittently connected mobile networks", In Proceedings of ACMSIGCOMM workshop on Delay Tolerant Networking (WDTN'5), pp 252-259, 2005.
[8] J. Burgess, B. Gallagher, D. Jensen and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," Proceedings of 25th IEEE International Conference on Computer Communications, Barcelona, 23-29 April 2006, pp. 1-11. doi:10.1109/INFOCOM.2006. 228
[9] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledgebased opportunistic forwarding in vehicular wireless ad hoc networks," In IEEE Vehicular Technology Conference (VTC), pp. 2289–2293, May 2005.
[10] J. Leguay, T. Friedman, V. Conan, "DTN Routing in a Mobility Pattern Space", presented at ACM SIGCOMM Workshop on Delay Tolerant Networking, 2005
[11] Hui, P. and Crowcroft, J. (2007) "Bubble rap: forwarding in small world dtns in every decreasing circles", Technical report, Technical Report UCAM-CL-TR684. Cambridge, UK: University of Cambridge.
[12] Boldrini, C., Conti, M., Jacopini, I., & Passarella, A.(2007, June). "HiBOp: A History Based Routing Protocol for Opportunistic Networks". Paper presented in the Proceedings of the WoWMoM 2007, Helsinki.
[13] Hemal Shah and Yogeshwar. P. Kosta , "Routing Enhancement Specific to Mobile Environment Using DTN", International Journal of Computer Theory and Engineering, Vol. 3, No. 4, August 2011
[14] T. Spyropoulos K. Psounis, C. S. Raghavendra "Efficient routing in intermittently connected mobile networks" The multiple copy case IEEE\ACM Trans. on Networking, Volume. 16, 2008.
[15] Wang, Guizhu, Bingting Wang, and Yongzhi Gao. "Dynamic spray and wait routing algorithm with quality of node in delay tolerant network."Communications and Mobile Computing (CMC), International Conference on. Volume. 3. IEEE, 2010.