An Algorithm for Automatic Web-Page Clustering using Link Structures

Debajyoti Mukhopadhyay and Sanasam Ranbir Sing

Abstract **- Web contains a large collection of heterogeneous documents. As a result, finding set of related pages from Web is currently facing one of the most crucial problems. The low precision Web search engines like Excite, Alta Vista etc. coupled with the ranked list presentation make it harder for users to find the information they are looking for. In this paper, we have proposed a methodology to cluster related pages using co-citations without manual study and/or predefined categories. These clusters are used to classify random pages in the Universe.**

Keywords **- Clustering, tightly-coupled, loosely-coupled, mixed-coupled, co-citation**.

1. INTRODUCTION

Several studies show Web search precision can be improved by clustering the Web documents. [2, 3, 4] Several automatic text classification methods [1, 7, 8, 9, 10], which use pre-specified classes of documents, play important role in finding information from Web resource. Typical, text-based classification methods utilize the words content in the documents as the most significant features. Unfortunately, a Web page might contain no obvious clues (textually) as to its intent [3].

Manual categorization of the pages like Yahoo! [13] provides higher precision categories. As millions of pages are accumulated in Web and the size of the Web is gradually increasing day-by-day, manual categorization is no more possible for large-scale web search engines. Most of the automatic Web categorization techniques assume to have predefined categories.

Flake et. al. [2] shows that Web pages form a community of their own. Several studies [3, 4, 5, 6,] find that link structure of the Web provide enough information to classify pages in the Web. In this paper, we have proposed a methodology to cluster related pages using co-citations without manual study and/or predefined categories. These clusters are used to classify random pages in the Universe. Very often, Web pages co-cite related pages. For example, personal home pages often co-cite *Pepsi* and *Coca-Cola* as their favorite soft drinks. Even though, their home pages does not provide obvious clue of their class, it is known that they belong to the same class.

In most of the commercial Web pages, pages in different categories are often co-cited. To avoid this ambiguity we have classified co-citation into three classes.

Tightly-Coupled: the pages are cited from one section/subsection i.e., under same category. For example, pages belonging to educational institutions, soft drinks, sports etc. are cited from personal home pages. Pages cited from same category are considered as tightly coupled co-citations. References given under scientific pages are taken as tightly coupled pages.

Loosely-Coupled: pages cited from different categories are considered as loosely-coupled co-citations.

Mixed-Coupled: Some pages do not have any category of their cited pages. Citations from such pages are considered as mixed-coupled co-citations.

We have considered tightly coupled co-citations only. Even in tightly coupled co-citations, we have found that some pages are not related. For example, in favorite category of a personal home page, movies, sports etc. are put together. To overcome this problem, we have used a normalization technique as explained in Section 2.2.

This paper has been organized as follows: Section 2 describes our approach; Section 3 provides the experimental setup and results; Section 4 has the concluding remarks.

2. PROPOSED METHODOLOGY

2.1 Co-citation and its normalized weight

Links are made by Web creators based on certain interest of their own. If the two pages are co-cited from some pages, then we consider that co-cited pages are related to each other for some reasons interested to Web creators. This relation defines an association between the two co-cited pages. As a reason, we have taken number of co-citations as the degree of association between the two co-cited pages.

We have defined co-citation weight for each co-cited pages as the total number of times being

Debajyoti Mukhopadhyay, Department of Computer Science & Engineering, St. Thomas' College of Engineering & Technology, 4 D.H. Road, Kolkata 700023, India. Email: debm@vsnl.com

Sanasam Ranbir Singh,Department of Computer Science & Engineering,Haldia Institute of Technology,P.O. Hatiberia, E. Midnapore, WB 721657, India. Email: san_ranbir@yahoo.com.

co-cited. The pages, which are published a long time back have got more number of co-citations than the pages which are published latter. As we are interested in recent popular pages as a base, we have normalized the co-citation weights in terms of number of co-citations per year.

Y(p,q) = (year of publication of page p or page q whichever is latter) – (current year).

Co(p,q) =Number of co-citation weight of page p and page q.

Nor_Co(p,q) = Co(p,q) / Y(p,q).

There are some pages, which are co-cited with fewer degree but fall in different categories. In order to remove such co-citations we have selected only the pages having normalized co-citation weight above a threshold T and defined a set S ={{p,q}U S | Nor_Co(p,q) ≥T and p, q Є Set in Universe}. Set S was initially empty and included all the page pairs, whose normalized co-citation weight is above threshold T. Set S has taken a Seed set and pages in S are clustered using the methods explained in Section 2.2.1and 2.2.2.

2.1.1. Co-Citation graph

Co-citation graph is an undirected graph. Each page in the set *S* represents a node in the co-citation graph. If two pages *p* and *q* are co-cited from some page, then they are connected by an edge.

From the graph, we can find out the disjoint sub-graphs using *Depth First Search* algorithm or *Breath First Search* algorithm. Each disjoint sub-graph is considered as a cluster of related pages. We have used the following algorithm to cluster the graph. The clusters obtained are considered as base cluster set.

Algorithm : Base_Cluster
Input : Graph(V, E)
Output : Set of clusters (C = { c_1, c_2, …., c_k})
Begin
while (V is not empty)
{ Create a new cluster c_i.
Get any v Є V and V=V - {v}.
Put v into c_i.
Include all the nodes which can be reached from v into c_i.
}

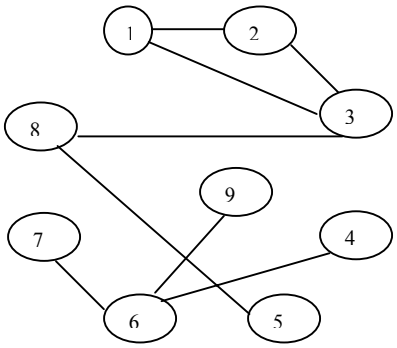

Figure 1: Co-citation graph of Example 1.

Example 1: Let us consider a graph as shown in the Figure1. In this graph, there are 9 nodes naming as 1,2,3 and so on up to 9. Two nodes are connected if there are co-cited from some pages. The edges are made randomly for the sake of explanation. Applying the algorithm explained above we find two cluster of pages {1, 2, 3, 5, 8} and {4, 6, 7, 9}.

2.1.2. Co-Citation table

When dealing with graphs having large number of nodes, it may be more convenient to record the co-citation information in form of a table. We have constructed a table called Co-Citation table to represent the graph. Each page represents a row/column as shown in table1. Pages are arranged in some order. If there are n pages and labeled as 1,2,3 and so on up to n representing 1st page, 2nd page and so on. Labels can be ordered as 1,2,3 and so on up to n. There are n-1 number of columns for 1st page to (n-1)th page arranging from left to right and n-1 number rows for 2nd page to nth page arranging from top to bottom. If two pages are connected i.e., they are co-cited in the set *S*, 1 is entered to their corresponding cell otherwise 0. The Co-Citation table of the Example 1 is shown in the Table1.

From the Co-Citation table, clusters are extracted using the following algorithm.

1. Start from the right most column of the table and proceed towards left until a column with non-zero cell is found. List all pages for which corresponding cell value is non-zero into one cluster.
2. Proceed to next column having at least one non-zero cell. Check whether this page has

co-citations with the pages in the previous clusters or not. If it has co-citations, then include this page and other pages having non-zero cell value in the same row into the cluster. Otherwise create a new cluster comprising of pages having non-zero cell value in the column.

3. Repeat step 2 until all the columns are processed.
4. The set of clusters obtained after step 3 is considered as the base cluster set.

Table1 shows the co-citation table of Example 1.

2	1							
3	1	1						
4	0	0	0					
5	0	0	0	0				
6	0	0	0	1	0			
7	0	0	0	0	0	1		
8	0	0	1	0	1	0	0	
9	0	0	0	0	0	1	0	0
	1	2	3	4	5	6	7	8

Table 1: Co-citation table of Example 1.

Clustering procedure:
Column 8: Φ
Column 7: Φ
Column 6: {6, 7, 9}
Column 5: {6, 7, 9}, {5, 8}
Column 4: {4, 6, 7, 9}, {5, 8}
Column 3: {4, 6, 7, 9}, {3, 5, 8}
Column 2: {4, 6, 7, 9}, {2, 3, 5, 8}
Column 1: {4, 6, 7, 9}, {1, 2, 3, 5, 8}

Base clusters for the graph of Example 1 are {4, 6, 7, 9}, {1, 2, 3, 5, 8}

2.1.3 Equivalent Partitions

Since we have constructed clusters of disjoint sub-graphs, we can define a relation between pages in the sub-graph.

Definition: The page p is said to have a relation with page q, pRq if there is a path from page p to page q in the co-citation graph.

By definition the relation R is a *reflexive relation*. Since the co-citation graph is an undirected graph, if there is path from page p to page q, then there is also a path from page q to page p. Thus, the relation R is a *symmetric relation* i.e., if pRq, then qRp. Again, if there is a path from page p to q and from page q to t, then there is a path from page p to t i.e., if pRq and qRt, then pRt. Thus, the relation R is a *transitive relation*. Hence, the relation R is an *equivalent relation* [14].

By the properties of an equivalence relation, we can construct a unique disjoint partition, Π. Π={1, 2, 3, 5, 8}, {3, 4, 7, 9} is the equivalence partition of the Example 1.

2.2 Clustering the Pages in Universe

In this Section, we have discussed a methodology to include a random page in Universe into a cluster. We have used the idea that a page has more number of both incoming and outgoing links within its cluster than outside the cluster. We have defined a cluster $C_{undefined}$ as the set of all page in Universe other than pages in S. $C_{undefined}$ = Set in Universe – S. The pages in $C_{undefined}$ are unmarked initially.

A random unmarked page p is taken from $C_{undefined}$ and marked. The probability of being into one of the other clusters in the based clusters set is calculated using the following formula.

$Pr(p, c_i)=L(p, c_i)/L(p)$ for all pages in the cluster $C_{undefined}$ and $c_i \varepsilon S U C_{undefined}$, where $Pr(p, c_i)$ is the probability of page p being in cluster c_i, $L(p, c_i)$ is the total number of incoming links to page p from c_i and outgoing links from page p to c_i and $L(p)$ is the total number of incoming and outgoing links to and from page p.

The page p is merged with cluster c_i, if c_i has highest number of incoming and outgoing links from and to page p than other clusters i.e., $c_i = \{c_i\} U \{p\}$ if $Pr(p, c_i)$ is maximum. Above process is repeatedly applied until no more unmarked pages is in $C_{undefined}$. The process is shown in algorithmic form below.

Algorithm : Clustering Random Pages
Input : Base Cluster and $C_{undefined}$
Output : Set of Clusters
Begin
Unmarked all the pages in $C_{undefined.}$
While (unmarked pages in $C_{undefined}$)
Select page $p \in C_{undefined}$ and marked.

For all cluster i

Calculate $Pr(p, c_i)=L(p, c_i)/L(p)$ where $Pr(p, c_i)$ is the probability of page p being in cluster c_i, $L(p, c_i)$ is the total number of incoming links from page p and outgoing links from page p to and from c_i and $L(p)$ is the total number of incoming and outgoing links to and from page p.

End of for

Select the cluster, c_i having highest $Pr(p, c_i)$

Merge p with c_i.

$C_{undefined} = \{C_{undefined}\} - \{p\}$.

End of while.

End.

Above algorithm explains only one scan over $C_{undefined}$ cluster. After each scan over $C_{undefined}$, it is highly possible that new pages are included in the base clusters. We can further merge pages in cluster $C_{undefined}$ with the clusters in base clusters as follows. Pages in $C_{undefined}$ are unmarked again and apply above algorithm repeatedly until no more pages can be merged with other clusters i.e., pages in the cluster $C_{undefined}$ are unchanged.

Once stable $C_{undefined}$ is found, we can conclude that pages in individual cluster of base clusters are related to each other under certain matrix. But we cannot say that pages in $C_{undefined}$ are related to each other. It may contain pages of different categories. If the cluster $C_{undefined}$ is not empty, we can further apply the processes of Section 2.1,and 2.2 on $C_{undefined}$ again and again taking $C_{undefined}$ as Universal set until significant number of clusters are obtained.

3. EXPERIMENTAL SETUP & RESULT

We have used the database of computer science related scientific literature created by CiteSeer [11], which is available at *http:// citeseer.nj.nec.com/directory.html*. CiteSeer is considered as the largest free full text scientific literature directory containing over 250,000 articles related to computer science. We have randomly collected around 3,500 articles on 25 different categories from CiteSeer directory as shown in Table 1. The articles are collected in the form of postscript. The postscript articles are converted to HTML form using PreScript [12], which is freely available. We have also collected around 300 random pages like "individuals' home page", "top institutions' home pages" etc.

The simulation program is implemented using C++ language and run on a Pentium II machine COMPAQ DESPRO with 64MB RAM, 40GB hard disk. We have fixed the threshold value to 25. There are around 560 co-cited pages above the threshold value and put into the base set S. The remaining pages which are not in the base set S are put into one cluster known as $C_{undefined}$.

Categories in CiteSeer Directory
World Wide Web
Agents
Electronic Communication
Metasearch
Search Engines
Architecture
Distributed Architecture
Mobile Agents
Parallel
Artificial Intelligence
Expert Systems
Knowledge Representation
Natural Language
Optimization
Machine Learning
Fuzzy Systems
Genetic Algorithm
Neural Networks
Information Retrieval
Classification
Extraction
Digital Libraries
Databases
Data Warehousing
Concurrency

Table 1: It shows the selected 25 categories from CiteSeer.

The pages in the base set S are clustered using the algorithm explained in Section 2.2.1 and found 17 base clusters. These clusters are taken as base clusters for clustering other random pages in $C_{undefined}$. Pages from the $C_{undefined}$ are taken one by one and merged with one of the 17 clusters using the algorithm given in Section 2.2. After applying the algorithm given in Section 2.2 again and again for 4 times, we have found that $C_{undefined}$ cluster is unchanged and found around 170 pages in $C_{undefined}$. Most of the institutional home pages fall in $C_{undefined}$.

We believe that applying the whole processes (Section 2) again on $C_{undefined}$ with smaller threshold value, $C_{undefined}$ can be further clustered. Since the

size of the $C_{undefined}$ is small, we have not clustered further. Including $C_{undefined}$, there are 18 clusters. Table 2 shows three cluster out of 18 clusters returned by our method. Table3 shows frequently found features in the clusters for three clusters shown in Table 2.

Cluster 1
The Anatomy of a Large-Scale Hypertextual Web Search Engine - Brin, Page
Authoritative Sources in a Hyperlinked Environment – Kleinberg
Querying Heterogeneous Information Sources Using Source Descriptions - Levy, Rajaraman, Ordille
Querying the World Wide Web - Mendelzon, Mihaila, Milo
Focused crawling: a new approach to topic-specific Web resource - Chakrabarti, van den Berg, Dom
Cluster 2
Mobile Ambients - Cardelli, Gordon
A calculus of mobile agents - Cédric Fournet, Georges Gonthier, Jean-Jacques Lévy, Luc Maranget, Didier Rémy
A calculus of mobile processes, parts 1-2. Information and Computation (context) - R. Milner, J. Parrow, and D. Walker
An asynchronous model of locality (context) - Amadio
The Architecture of the Ara Platform for Mobile Agents - Peine, Stolpmann
Cluster 3
KQML as an agent communication language - Finin, Labrou, Mayfield
A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text. - Joachims
Text Categorization with Support Vector Machines: Learning with Many - Joachims
Querying Semi-Structured Data - Abiteboul
An Evaluation of Statistical Approaches to Text Categorization - Yang

Table 2: It shows three clusters out of 18 clusters obtained from our system with some of the tops pages.

Cluster 1
World wide web, search engines, classification, page rank, hyperlink, authorities, hubs, citation …..
Cluster 2
Agents, mobile, mobile agents, processes, computing, mobile computing, distributed……
Cluster 3
Learning, machine learning, Text categorization, pattern recognition, training, models, fuzzy, neural, semi-structure, machines….

Table 3: It shows some of the top common features in each cluster given in Table 2.

We have found that some pages in different categories in CiteSeer categories are merged to same category. For example, the articles The Anatomy of a Large-Scale Hypertextual Web Search Engine - Brin, Page and The Structure of Broad Topics on the Web – Chakrabarti et. al., which are in different categories in CiteSeer are in Category 1. Similarly significant amount of pages in different pages are merged to same category.

In our method, clusters are non-overlapped to each other. Though we have collected articles on 25 different categories and 300 random pages, our approach returns only 18 clusters. It returns broader region.

We have observed that home pages like of Steve Lawrence's home page, Soumen Chakrabarti's home page are merged with cluster 1. It gives some clue about the personal interest.

4. CONCLUSION

In most of the cases, tightly couple co-cited pages (defined in Section 1) are related to each other under certain matrix. If the number of co-citation is very large, then we can assume that they are related to each other. The base clusters set returned by our system contains only the highly co-cited pages. Unlike machine learning based classification techniques [7, 9, 10], which assume to have predefined clusters, our system generate base clusters, which are found to be highly related to each other. In addition, traditional machine learning base classification techniques can be applied on top of base clusters return by our system.

5. REFERENCES

[1] James Tin-Yau Kwok. "Automated Text Categorization Using Support Vector Machine". In *proceedings of ICONIP'98, 5th International Conference on Neural Information Processing.*

[2] Gary William Flake, Steve Lawrence, C. Lee Giles, Frans M. Coetzee. "Self Organization and Identification of Web Communities". *IEEE Computer, 35(3), 66-71, 2000.*

[3] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, Gary W. Flake. "Using Web Structure for Classifying and Describing Web Pages," *WWW2002, Honolulu, Hawaii, USA, 7-11 May 2002.*

[4] Soumen Chakrabarti, Byron E. Dom, Ravi Kumar, Prabhakar Raghavan, Shidhar Rajagopalan, Andrew Tomkins, David Gibson, Jon Kleinberg. "Mining the Web's Link Structure," *IEEE Computer, (32)8: August 1999, pp 60-67.*

[5] B. D. Davison. "Topical Locality in the Web". In *Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval (SIGIR 2000)*, ACM, Athens, Greece, July 2000. pp 272-279.

[6] J. Furnkranz. Exploiting Structure Information for text Classification on the WWW. In *Intelligent Data Analysis, 1999, pp. 487-498.*

[7] Andrew McCallum and Kamal Nigam. "A Comparison of Event Models for Naive Bayes Text Classification". In *AAAI/ICML-98 Workshop on Learning for Text Categorization*, pp. 41-48.

[8] Y. Yang and X. Liu. "Re-examination of text categorization methods". In *SIGIR-99, 1999.*

[9] Hwanjo Yu, Jiawei Han and Kevin Chen-Chuan Chang. PEBL: "Positive Example Based Learning for Web Page Classification Using SVM". In *SIGKDD'02 Edmonton, Alberta, Canada.*

[10] C. Apte and F. Damerau. "Automated learning of decision rules for text categorization". In *ACM Transactions on Information System, 12(3):233-251, July 1994.*

[11] CiteSeer. http://citeseer.nj.nec.com

[12]PreScript: PostScript to Text. www.nzdl.org/html/prescript.html

[13] Yahoo!. Yahoo! search engine. *http://www.yahoo.com*

[14] J.P. Tremblay, R. Manohar, "Discrete Mathematical Structures with Applications to Computer Science". McGraw-Hill.