

# **'Web Rings' and Tree structures on the web**

**By**  
**Maurice T. Mouawad**

**A Thesis**

Submitted in Partial Fulfillment of the  
Requirements for the Degree of Master of  
Science in Computer Science

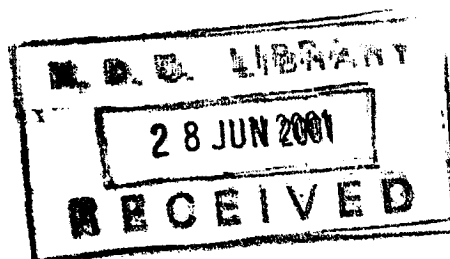
Department of Computer Science

Faculty of Natural and Applied Science

Notre Dame University- Louaize

Zouk Mousbeh, Lebanon

June 2001

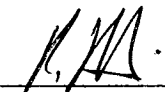


# 'Web Rings' and Tree structure on the web

By

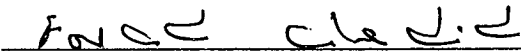
**Maurice T. Mouawad**

Approved



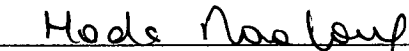
---

Khalidoun El-Khalidi: Assistant professor of Computer Science.  
Advisor.



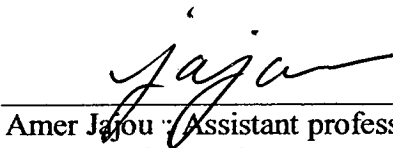
---

Fouad Chedid: Associate professor of Computer Science and Chairperson.  
Member of committee.



---

Hoda Maalouf: Assistant professor of Computer Science.  
Member of committee



---

Amer Jajou: Assistant professor of Mathematics.  
Member of committee.

Date of thesis defense: June 19, 2001

## **ACKNOWLEDGEMENTS**

First, I would like to thank my Advisor Dr. Khaldoun El-Khalidi for his consistent support, and for being helpful in encouraging me and providing me with the correct information throughout a whole year in my studies and research.

Second, I would like to thank Dr Fouad Chedid, Chairperson of Computer Science department, for being supportive and generous throughout the whole period of my Master degree.

Also, I am thankful for all the Doctors who provided me with valuable information throughout my education period.. I would like to express my deep feelings for the university that gave me the chance to move one step forward in my education, Notre Dame University. May God gives it and its employees a long lasting life .

Finally I would like to thank my family for the love and encouragement they offered me through out my whole life.

## ABSTRACT

The web is a vast source of information. However, due to the divisiveness and unlikeness of web pages' contents, this information is covered in the chaotic structure of the World Wide Web. At the same time, with the spread of web access, search engines are being, if not the sole utility, one of the most mechanisms used by the increasing number of users, to find interesting information. We are interested in identifying how pieces of information represented by URL pages, sharing common topics, are related as they are represented on the web. One such problem is studying patterns of occurrences of gathered linked pages named communities and their structure in the web. We call this the web design structure problem. Trying to identify different structures of newly emerging communities materializes the structure problem: communities that have little or no dense representation. More over, we will try to find shapes that communities can take. This case study analysis is based on graph-theory and some optimization approaches, which will help in the algorithmic engineering necessary for description of simple, iterative, and optimized algorithms to find community structures. Also, it will include implementations of these algorithms based on a data source extracted manually from the web, an analysis of the results obtained and some ideas for further work

## Table of contents

List Of Figures	vii
List Of tables	viii
1.Introduction	1
2.Overview	3
2.1.Introduction	3
2.2.Graph structure in the web	3
2.2.1.A brief on graph theory and terminology	3
2.2.2.Experimental results	5
2.2.3.Related work	8
2.2.4.Graph theoretic methods	9
2.3.Mining the web for communities	9
2.3.1.H.I.T.S (Hyperlinked-Induced Topic Search)	10
2.3.1.1.Focused subgraph	11
2.3.1.2.Computing hubs and authorities	11
2.3.1.3.H.I.T.S	12
2.3.1.4.Diffusion and generalization	14
2.3.2.Trawling the web for emerging cyber communities	15
2.3.2.1.Bipartite cores	15
3.Implementation of H.I.T.S	19
3.1.Introduction	19
3.2.Data collection	19
3.2.1.The search engine Google	19
3.2.2.Construction of the root set $R_\sigma$	20
3.2.3.Construction of the base set $S_\sigma$	22
3.2.4.Data structure	22
3.2.5.Computing hubs and authorities	23

4.Implementation of trawling	28
4.1.Introduction	28
4.2.Optimization	28
4.3.Trawling	28
5.Tree structure	32
5.1.Introduction	32
5.2.Tree structure	32
5.2.1.Tree generation	34
5.2.2.Implementation	35
6.Webring structure	36
6.1.Introduction	36
6.2.Webrings	36
6.2.1.Advantages of webrings	37
6.2.2.Virtual shape	37
6.2.3.Enumerating webrings	38
6.2.3.1.Data source collection	39
6.2.3.2.Optimization	39
6.2.3.4.Ring algorithm	41
6.3.Implementation	43
7.Conclusion	45
References	50

## LIST OF FIGURES

Figure	
1.Connectivity of the web	7
2.Pages referenced together	12
3.Core $C(1,4)$	30
4.Core(4,1)	31
5.Core(2,3)	31
6.Binary tree	33
7.core $c(i, j)$	33
8.Bipartite tree	34
9.Resulting tree	35
10.Webring	36
11.Virtual segment	37
12.Virtual triangle	37
13.Virtual tetrad	38
14.Resulting ring tree	41

## LIST OF TABLES

Table	
1. Authority comparison for $k=1$	23
2. Hub comparison for $k=1$	24
3. Authority comparison for $k=5$	24
4. Hub comparison for $k=5$	25
5. Authority comparison for $k=9$	25
6. Hub comparison for $k=9$	26
7. Bipartite core list	30
8. Links table	43



# Chapter 1

## Introduction

The analysis of an hyperlinked environment structure is a task with many facets- its precise goals depend deeply on the nature of the environment case study, in this situation it is the World Wide Web. The problem of identifying meaningful structures is compelling for several reasons: the WWW is a hypertext corpus of enormous complexity, and it continues to grow at a phenomenal rate. Moreover, it can be viewed as a puzzling form of " populist hypermedia", in which millions of online participants, many with conflicting goals, are continuously creating hyperlinked content. Thus, while individuals can impose structure at an extremely local level, its global organization is completely "unplanned"- in some sense, high-level structure can emerge only through a delayed analysis. Our emphasis here is on an investigation of the link topology of the WWW. The World Wide Web has several thousand well-known, explicitly defined communities: groups of individuals who share common interest, together with the web pages most popular amongst them. These communities are easy to find it is simply a matter of visiting the appropriate portal or newsgroup. On the other hand, the chaotic nature of the World Wide Web has resulted in many more implicitly-defined communities. These communities are far from being recognized by famous webportals such as Altavista or Yahoo

The subject of this thesis is to study the nature and structure of these newly generated communities by analyzing the structure of these communities' cores or in some sort nuclei. There are several reasons for extracting newly emerging communities, mainly because these communities provide valuable and possibly the most reliable, timely, and up-to-date information resources for a user interested in them, and because they represent the sociology of the web: studying them gives insights into the intellectual evolution of the web. On the other hand an analysis of the structure of the cores of these communities will concentrate on tree and webring structure. The tree structure is beneficial at least for two reasons: space and time optimization for web portal's database where each tree can

be represented by its root and the nodes and leaves will be parsed. For the webring structure, mainly it will optimize the results of search engine queries.

Chapter 2 is an overview about communities, relations and graphs. It will include some algorithmic's description used for extracting communities. Chapter 3 will focus on the implementation of HITS. Chapter 4 will implement Trawling. Chapter 5 will include search for tree structures and the implementation of the tree algorithm . Chap 6 will discuss the web ring structure and it will include an implementation and the corresponding results.

## Chapter 2

### Overview

#### 2.1 Introduction

The study of the web as a graph is not only fascinating in its own right, but also yields valuable insight into web algorithms for crawling, searching and community discovery, and the sociological phenomena which characterize its evolution

Consider the directed graph whose nodes correspond to static pages on the web, and whose arcs correspond to hyperlinks between these pages. We study various properties of this graph including its diameter, degree distributions, connected components, and macroscopic structure. There are several reasons for developing an understanding of this graph:

1. Designing crawl strategies on the web.
2. Understanding of the sociology of content creation on the web.
3. Analyzing the behavior of web algorithms that make use of link information. To take just one example, what can be said of the distribution and evolution of PageRank [6] values on graphs like the web?
4. Predicting the evolution of web structures such as bipartite cores [4] and 'Web Rings', and better algorithms for discovering and organizing them.
5. Predicting the emergence of new, yet unexploited phenomena in the web graph.

#### 2.2 Graph structure in the web

Before going deeper in the graph structure and some experiments proving some laws it would be better if a brief on graph theory terminology is included.

##### 2.2.1 A brief on graph theory and terminology

A directed graph consists of a set of nodes, denoted  $V$  and a set of arcs, denoted  $E$ . Each arc is an ordered pair of nodes  $(u,v)$  representing a directed connection from  $u$  to  $v$ . The out-degree of a node  $u$  is the number of distinct arcs  $(u,v_1)\dots(u,v_k)$  (i.e., the number of links from  $u$ ), and the in-degree is the number of distinct arcs  $(v_1,u)\dots(v_k,u)$  (i.e., the

number of links to  $u$ ). A path from node  $u$  to node  $v$  is a sequence of arcs  $(u, u_1)$ ,  $(u_1, u_2)$ , ...  $(u_k, v)$ . One can follow such a sequence of arcs to "walk" through the graph from  $u$  to  $v$ . Note that a path from  $u$  to  $v$  does not imply a path from  $v$  to  $u$ . The distance from  $u$  to  $v$  is one more than the smallest  $k$  for which such a path exists. If no path exists, the distance from  $u$  to  $v$  is defined to be infinity. If  $(u, v)$  is an arc, then the distance from  $u$  to  $v$  is 1.

Given a directed graph, a strongly connected component (strong component for brevity) of this graph is a set of nodes such that for any pair of nodes  $u$  and  $v$  in the set there is a path from  $u$  to  $v$ . In general, a directed graph may have one or many strong components. The strong components of a graph consist of disjoint sets of nodes.

An undirected graph consists of a set of nodes and a set of edges, each of which is an unordered pair  $\{u, v\}$  of nodes. In our context, we say there is an edge between  $u$  and  $v$  if there is a hyperlink between  $u$  and  $v$ , without regard to whether the link points from  $u$  to  $v$  or the other way around. The degree of a node  $u$  is the number of edges incident to  $u$ . A path is defined as for directed graphs, except that now the existence of a path from  $u$  to  $v$  implies a path from  $v$  to  $u$ . A component of an undirected graph is a set of nodes such that for any pair of nodes  $u$  and  $v$  in the set there is a path from  $u$  to  $v$ . We refer to the components of the undirected graph obtained from a directed graph by ignoring the directions of its arcs as the weak components of the directed graph. Thus two nodes on the web may be in the same weak component even though there is no directed path between them (consider, for instance, a node  $u$  that points to two other nodes  $v$  and  $w$ ; then  $v$  and  $w$  are in the same weak component even though there may be no sequence of links leading from  $v$  to  $w$  or vice versa). The interplay of strong and weak components on the (directed) web graph turns out to reveal some unexpected properties of the web's connectivity.

A breadth-first search (BFS) on a directed graph begins at a node  $u$  of the graph, and proceeds to build up the set of nodes reachable from  $u$  in a series of layers. Layer 1 consists of all nodes that are pointed to by an arc from  $u$ . Layer  $k$  consists of all nodes to which there is an arc from some vertex in layer  $k-1$ , but are not in any earlier layer. Notice that by definition, layers are disjoint. The distance of any node from  $u$  can be read

out of the breadth-first search. The shortest path from  $u$  to  $v$  is the index of the layer  $v$  belongs in -- if there is such a layer. On the other hand, note that a node that cannot be reached from  $u$  does not belong in any layer, and thus we define the distance to be infinity. A BFS on an undirected graph is defined analogously.

Finally, the diameter of a graph, directed or undirected, is the maximum over all ordered pairs  $(u,v)$  of the shortest path from  $u$  to  $v$ . Some researchers have proposed studying the average distance of a graph, defined to be the length of the shortest path from  $u$  to  $v$ , averaged over all ordered pairs  $(u,v)$ ; this is referred to as diameter. The difficulty with this notion is that even a single pair  $(u,v)$  with no path from  $u$  to  $v$  results in an infinite average distance. This motivates the following revised definition: let  $P$  be the set of all ordered pairs  $(u,v)$  such that there is a path from  $u$  to  $v$ . The average connected distance is the expected length of the shortest path, where the expectation is over uniform choices from  $P$ .

### 2.2.2 Experimental results

In their work [1], they tried to elaborate a study on graph structure depending on some experiments which are mentioned here as an indication to the important results they obtained. They detail a number of experiments on a web crawl of approximately 200 million pages and 1.5 billion hyperlinks; the scale of this experiment is thus five times larger than the previous biggest study [4] of structural properties of the web graph, which used a pruned data set from 1997 containing about 40 million pages. Recent work ([4] on the 1997 crawl, and [3] on the approximately 325K-node nd.edu subset of the web) has suggested that the distribution of degrees (especially in-degrees) follows a power law:

The power law for in degree: the probability that a node has in-degree  $i$  is proportional to  $1/i^x$ , for some positive  $x > 1$ .

They verify the power law phenomenon in current (considerably larger) web crawls, confirming it as a basic web property.

In other recent work, [2] report the intriguing finding that most pairs of pages on the web are separated by a handful of links, almost always under 20, and suggest that this

number will grow logarithmically with the size of the web. This is viewed by some as a "small world" phenomenon. Their experimental evidence reveals a rather more detailed and subtle picture: significant portions of the web cannot at all be reached from other (significant) portions of the web, and there is significant number of pairs that can be bridged, but only using paths going through hundreds of intermediate pages.

They performed three sets of experiments on web crawls from May 1999 and October 1999. First, they generated the in and out-degree distributions, confirming previous reports on power laws; for instance, the fraction of web pages with  $i$  in links is proportional to  $1/i^{2.1}$ , the constant 2.1 being in remarkable agreement with earlier studies at varying scales [2,3]. In their second set of experiments they studied the directed and undirected connected components of the web. They show that power laws also arise in the distribution of sizes of these connected components. Finally, they performed a number of breadth-first searches from randomly chosen start nodes.

Their analysis reveals an interesting picture (figure1) of the web's macroscopic structure. Most (over 90%) of the approximately 203 million nodes in the crawl they performed form a single connected component if hyperlinks are treated as undirected edges. This connected web breaks naturally into four pieces. The first piece is a central core, all of whose pages can reach one another along directed hyperlinks -- this "giant strongly connected component" --(SCC) is at the heart of the web. The second and third pieces are called IN and OUT. IN consists of pages that can reach the SCC, but cannot be reached from it possibly new sites that people have not yet discovered and linked to. OUT consists of pages that are accessible from the SCC, but do not link back to it, such as corporate web sites that contain only internal links. Finally, the TENDRILS contain pages that cannot reach the SCC, and cannot be reached from the SCC. Perhaps the most surprising fact is that the size of the SCC is relatively small -- it comprises about 56M pages. Each of the other three sets contain about 44M pages -- thus, all four sets have roughly the same size.

They show that the diameter of the central core (SCC) is at least 28, and that the diameter of the graph as a whole is over 500. They show that for randomly chosen source and destination pages, the probability that any path exists from the source to the destination is only 24%. They also show that, if a directed path exists, its average length will be about

16. Likewise, if an undirected path exists (i.e., links can be followed forwards or backwards), its average length will be about 6. These results are remarkably consistent across two different, large AltaVista crawls. This suggests that their results are relatively insensitive to the particular crawl they use, provided it is large enough.

In a sense the web is much like a complicated organism, in which the local structure in a microscopic scale looks very regular like a biological cell, but the global structure exhibits interesting morphological structure (body and limbs) that are not obviously evident in the local structure. Therefore, while it might be tempting to draw conclusions about the structure of the web graph from a local picture of it, such conclusions may be misleading.

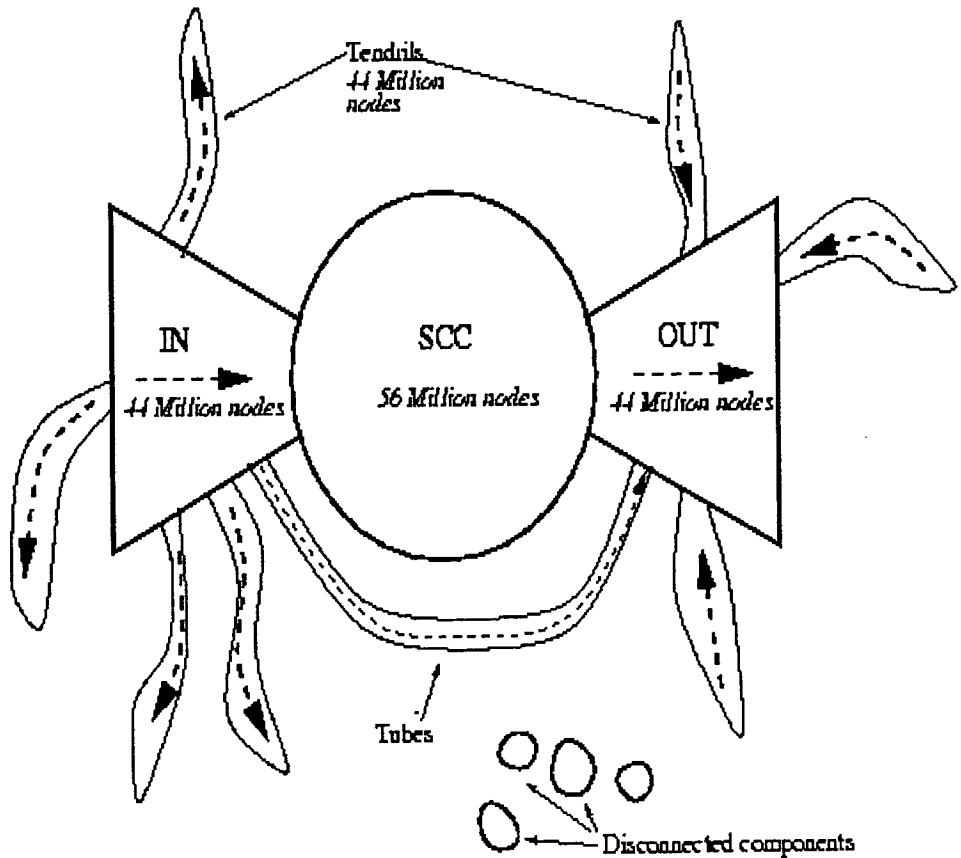


Figure 1: Connectivity of the web: one can pass from any node of IN through SCC to any node of OUT. Hanging off IN and OUT are TENDRILS containing nodes that are reachable from portions of IN, or that can reach portions of OUT, without passage through SCC. It is possible for a TENDRIL hanging off from IN to be hooked into a TENDRIL leading into OUT, forming a TUBE – a passage from a portion of IN to a portion of OUT without touching SCC.

### 2.2.3 Related work

Broadly speaking, related work can be classified into two groups:

- (1) Observations of the power law distributions on the web; and
- (2) Work on applying graph theoretic methods to the web.

#### **Zipf-Pareto-Yule and Power laws.**

Distributions with an inverse polynomial tail have been observed in a number of contexts. The earliest observations are due to Pareto [3] in the context of economic models. Subsequently, these statistical behaviors have been observed in the context of literary vocabulary [15], sociological models [3], and even oligonucleotide sequences [4] among others. In [1], their focus is on the closely related power law distributions, defined on the positive integers, with the probability of the value  $i$  being proportional to  $1/i^k$  for a small positive number  $k$ . Perhaps the first rigorous effort to define and analyze a model for power law distributions is due to Herbert Simon [5].

More recently, power law distributions have been observed in various aspects of the web. Two lines of work are of particular interest. First, power laws have been found to characterize user behavior on the web in two related but dual forms:

1. access statistics for web pages, which can be easily obtained from server logs (but for caching effects);
2. the number of times users at a single site access particular pages also enjoy power laws, as verified by instrumenting and inspecting logs from web caches, proxies, and clients.

Second, and more relevant to our immediate context is the distribution of degrees on the web graph. In this context, recent work [3,8] suggests that both the in- and the out-degrees of vertices on the web graph have power laws. The difference in scope in these two experiments is noteworthy. The first [4] examines a web crawl from 1997 due to Alexa, Inc., with a total of over 40 million nodes. The second [3], examines web pages from the University of Notre Dame domain, \*.ndu.edu, as well as a portion of the web reachable from 3 other URLs. This collection of findings reveals an almost fractal like quality for the power law in-degree and out-degree distributions, in that it appears both as



a macroscopic phenomenon on the entire web, as a microscopic phenomenon at the level of a single university web site, and at intermediate levels between these two.

There is no evidence that users' browsing behavior, access statistics and the linkage statistics on the web graph are related in any fundamental way, although it is very tempting to conjecture that this is indeed the case. It is usually the case, though not always so, that pages with high in-degree will also have high PageRank [6]. Indeed, one way of viewing PageRank is that it puts a number on how easy (or difficult) it is to find particular pages by a browsing-like activity. Consequently, it is plausible that the in-degree distributions induce a similar distribution on browsing activity and consequently, on access statistics.

#### **2.2.4 Graph theoretic methods.**

Much recent work has addressed the web as a graph and applied algorithmic methods from graph theory in addressing a slew of search, retrieval, and mining problems on the web. The efficacy of these methods was already evident even in early local expansion techniques. Since then, the increasing sophistication of the techniques used, the incorporation of graph theoretical methods with both classical and new methods which examine context and content, and richer browsing paradigms have enhanced and validated the study and use of such methods. Following, the view that connected and strongly connected components represent meaningful entities has become accepted. [9] augment graph theoretic analysis to include document content, as well as usage statistics, resulting in a rich understanding of domain structure and a taxonomy of roles played by web pages.

Graph theoretic methods have been used for search [10,7,4,15], browsing and information foraging, and web mining.

### **2.3 Mining the web for communities**

The World Wide Web is growing in an anarchic pattern. Millions of pages are being created randomly without the minimum requirements of organization. Creation of web pages leads to another important problem: the increasing number of hyperlinks, since for each page created there will be several links created to and from this same page.

Surprisingly, opposite to what we think, these pages while being created randomly and chaotically they are being assembled in-groups called communities. One of the main interests is to study the structure of these communities. Jon Kleinberg addressed this issue in [11] by elaborating an interesting algorithm called HITS (Hyperlink-Induced Topic Search). On the other hand [9] tried to enumerate the communities that are implicitly formed, and the web portals could not detect them. But in [12] R.Lempel and S.Moran created another algorithm based on the same principle as the HITS' but with some variations in the graph theory which in their opinion are necessary for the tightly knight communities to be identified and which HITS does not do.

### **2.3.1 H.I.T.S ( Hyperlinked-Induced Topic Search)**

The world wide web with it's enormous complexity has imposed an interesting problem, for a while it has been solved using web portals search engine: users interested in a specific topic used to submit a query to a search engine hoping to get a significant result. But with the growth of the WWW, discovering pages using regular web portals became difficult, especially with the abundance of web pages relevant in one way or another to the topic in question. The relevance of a subject depends mainly on the human evaluation, which does not give solid basis for a regular search. Jon Kleinberg in [11] talked about three kinds of queries: specific queries, broad topic queries, similar page queries. For the first one, the main obstacle is the scarcity of web pages relevant to the topic, on the other hand, the second kind of queries offers a big number of choices such that it becomes an abundance problem. For this reason, he elaborated the notion of authoritative pages that can be used from an effective search. He depended mainly on link structure to find the most authoritative pages based on the idea that if a page  $p$  has a link to page  $q$  then in some measure  $p$  has conferred authority on  $q$ . Moreover, links afford us the opportunity to find potential authorities purely through the pages that point to them. His model for effective search is based on the relationship that exists between the authorities for a topic and those pages that link to many related authorities, he referred to pages of this latter type as hubs [11]. Since the entire set of pages relevant to a broad topic query can have a size in the millions, Jon Kleinberg used a focused sub graph of the WWW to extract the most authoritative pages for a given topic.

### 2.3.1.1 Focused sub graph

The construction of the focused sub graph passes through two main steps: the collecting of the root set  $R_\sigma$  and the foundation of the base set  $S_\sigma$  where  $\sigma$  represents the query string. For the formation of the root set, first apply a string query to a web portal such as Altavista and collect the highest-ranked pages. This root set satisfies two criterion:

- 1 the collection is relatively small
- 2 The collection is rich in relevant pages

but the root set does not satisfy the criteria that it should contain most of the strongest authorities especially that all the links in  $R_\sigma$  are structureless . Therefor  $R_\sigma$  is grown into a bigger set, the base set, by finding the out-links and in-links of all the pages in  $R_\sigma$ . Thus  $S_\sigma$  is obtained by growing  $R_\sigma$  to include any page pointed to by a page in  $R_\sigma$  and any page that points to a page in  $R_\sigma$ . With the restriction that a single page in  $R_\sigma$  brings at most  $d$  pages pointing to it into  $S_\sigma$ . There are two kinds of links in  $S_\sigma$ , intrinsic and transverse. A transverse link is a link between two different domains. This kind is kept. On the other hand, intrinsic links, are links between the same domain and are used purely for navigation reasons. These links are already deleted .

### 2.3.1.2 computing hubs and authorities

After the construction of the base set, which includes a good number of relevant authoritative pages, there should be a way to extract these pages from between all the others remaining in  $S_\sigma$  . One could think of ordering all pages in  $S_\sigma$  by their in-degree; the number of links that point to each page. This methodology is necessary but not enough because there exists several pages which are highly referenced by other ones just for advertising reasons, even though it is irrelevant to the topic in concern. In [11] Jon Kleinberg circumvented this problem using the following observation: Authoritative pages relevant to the initial query should not only have large in-degree; since they are all authorities on a common topic, there should also be considerable overlap in the sets of pages that point to them. Thus in addition to highly authoritative pages, we expect to find what could be called hub pages: these are pages that have links to multiple relevant

authoritative pages. It is these hub pages that pull together authorities on a common topic, and allow us to throw out unrelated pages of large in-degree.

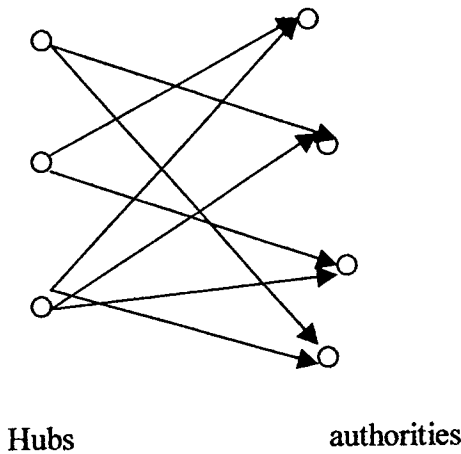


Figure 2 : pages referenced together

Hubs and authorities exhibit what could be called a mutually reinforcing relationship: a good hub is a page that points to many good authorities, a good authority is a page that is pointed to by many good hubs.

### 2.3.1.3 H.I.T.S

To break the mutually reinforcing circularity, Kleinberg used an iterative algorithm that appends and updates numerical authority  $x^p$  and hub  $y^p$  weight for each page  $p$ . The weights are positive and normalized so that

$$\sum_{p \in S_\sigma} (x^p)^2 = 1 \quad \text{and} \quad \sum_{p \in S_\sigma} (y^p)^2 = 1$$

The pages with large  $x$ -values are authority pages.

The pages with large  $y$ -values are hub pages.

But the mutually reinforcing relationship is not represented yet. It is expressed in the algorithm as follows:

If page  $p$  points to many pages with large  $x$ -values, then it should receive a large  $y$ -value, and if page  $p$  is pointed to by many pages with large  $y$ -values then it should receive a large  $x$ -value.

Practically, in the algorithm, there will be two operations I and O.:

Given the set of weights  $\{ x^p \}$  and  $\{ y^q \}$

The I operation updates the x-weights as follows:

$$x^p \xleftarrow{\Sigma_q} \xrightarrow{p} y^q$$

The O operation updates the y-weights as follows:

$$y^p \xleftarrow{\Sigma_p} \xrightarrow{q} x^q$$

In the algorithm, there are two vectors  $x_0$  and  $y_0$  initialized to 1 each. These two vectors represent the x-values and the y-values for all pages in  $S_\sigma$ . Then the I and O operations are applied mutually for K iterations:

I is applied to  $(x_{i-1}, y_{i-1})$  getting  $x_i'$

O is applied to  $(x_i', y_{i-1})$  getting  $y_i'$

And then the top C authorities and hubs are output, usually  $C = 5$  to 10.

For K, the number of iterations, Kleinberg showed that as one applies iteration with large values of K, the sequences of vectors  $\{x_k\}$  and  $\{y_k\}$  converge to fixed points  $x^*$  and  $y^*$ . He depended on the two theorems which are mentioned below to prove that  $x^*$  and  $y^*$  are the principal eigenvector of  $A^T A$  and  $AA^T$ .

**Theorem1:** The sequences  $x_1, x_2, x_3, \dots$  and  $y_1, y_2, y_3, \dots$  converge to limits  $x^*$  and  $y^*$  respectively.

**Theorem2:**  $x^*$  is the eigenvector of  $A^T A$  and  $y^*$  is the principal eigenvector of  $AA^T$

At the end of this brief explanation of Kleinberg's algorithm, it is worth mentioning that the use of textual content of pages was only at the beginning with the query using web portals which produced the root set  $R_\sigma$ .

Eventhough H.I.T.S produces the top authority and hub pages concerning a topic, there are some topics that may have several very different meanings E.g "jaguar". This diversion of meanings will lead, when a query is applied, to several collections of pages

all relevant to the topic but each collection is separated from the other ones. One could be interested in finding several densely linked collections of hubs and authorities among the same base set  $S_\sigma$  of pages. In H.I.T.S Kleinberg related the hubs and authorities computed by the algorithm to the principal eigenvectors of the matrices  $A^T A$  and  $AA^T$  where  $A$  is the adjacency matrix of  $S_\sigma$ . But the non principal eigenvectors of  $A^T A$  and  $AA^T$  offers a natural way to extract additional densely linked collections of hubs and authorities from the base set  $S_\sigma$ . Kleinberg used the following fact:

$A^T A$  and  $AA^T$  have the same multi-set of eigenvalues, and their eigenvectors can be chosen so that  $W_i(AA^T) = W_i(A^T A)$ .

Thus the I and O operations mentioned earlier, applied to the eigenvectors  $x_i^* = W_i(A^T A)$  and  $y_i^* = W_i(AA^T)$ , keep the x-weights and the y-weights parallel to  $x_i^*$  and  $y_i^*$ . Therefore each pair of weights  $(x_i^*, y_i^*)$  has precisely the mutually relationship needed in authority/hub pairs. Moreover, applying I/O operations multiplies the magnitudes of  $x_i^*$  by a factor of  $|\lambda_i|$ ; thus  $|\lambda_i|$  gives precisely the extent to which the hub weight  $y_i^*$  and authority weight  $x_i^*$  reinforce one another.

#### 2.3.1.4 Diffusion and generalization

Kleinberg depends mainly on links between pages to compute a densely linked collection of pages without regard to their contents. How much relevant these pages are, depends mainly on the way the base set is constructed. Kleinberg tried to ensure that the base set includes many relevant authoritative and hub pages. But the problem occurs when the query string specifies a topic not sufficiently broad. This means, there will not be enough relevant pages in the base set from which to extract a sufficiently dense subgraph of relevant hubs and authorities. As a result, authoritative pages corresponding to competing "broader" topics will win out over the pages relevant to the query string, and be returned by the algorithm. In such case the process has diffused from the initial query. It is interesting that too specific query string  $\sigma$  very often represents a natural generalization of  $\sigma$ .

### **2.3.2 Trawling the web for emerging cyber-communities**

The subject of trawling is the identification of implicitly-defined communities [9], communities that does not manifest as resource collections in directories such as Yahoo! and Infoseek. The reasons for extracting such communities from the web as they emerge are important. First, these communities provide valuable and possibly the most reliable, timely and up-to-date information resources for a user interested in them. Second, they represent the sociology of the web: studying them gives insights of the intellectual evolution of the web. Finally web portals identifying and distinguishing between these communities can target advertising at a very precise level. [9] depends, on finding structures that are relatively rare, on the co-citation relationship which is effectively the join of the web "points to" relation and its transposed version, the web "pointed to by" relation.

#### **2.3.2.1 Bipartite cores**

The authors of [9] started with the basic idea that web sites that should be part of the same community frequently do not reference one another for competitive reasons or because the sites do not share a point of view. Linkage between these related pages can be established by a phenomenon that occurs repeatedly: co-citation. Co-citation is a concept, which originated in the bibliometrics' literature [5]. The main idea is that pages that are related are frequently referenced together. This assertion is truer on the web where linking is an essential navigational element. Thus, co-citation can be an early indicator of newly emerging communities, communities that have taken shape even before the participants have realized that they have formed a community.

[9] developed a mathematical intuition which states that web communities are characterized by dense directed bipartite sub-graphs. A bipartite sub-graph, is a graph, whose node set can be partitioned into two sets  $F$  and  $C$ . Every directed edge in the graph is directed from a node  $u$  in  $F$  to a node  $v$  in  $C$ . A bipartite graph is dense if many of the possible edges between  $F$  and  $C$  are present. The dense bipartite graphs that are signature of web communities contain at least one core, where a core is a complete bipartite sub-

graph with at least  $i$  nodes from  $F$  and  $j$  nodes from  $C$ . Recall that a complete bipartite graph on node-sets  $F$  and  $C$ , contains all possible edges between a vertex of  $F$  and a vertex of  $C$ . Thus, the core is a small  $(i, j)$ -sized complete bipartite sub-graph of the community. A community can be found by finding first its core, and then use the core to find the rest of the community.

A fact about random bipartite graphs states that:

Let  $B$  be a random bipartite graph with edges directed from a set  $L$  of nodes to a set  $R$  of nodes, with  $m$  random edges each placed between a vertex of  $L$  and a vertex of  $R$  uniformly at random, then there exists  $i, j$  that are functions of  $(|L|, |R|, m)$  such that with high probability,  $B$  contains  $i$  nodes from  $L$  and  $j$  nodes from  $R$  forming a complete bipartite sub-graph.

Based on this fact, [9] elaborated a hypothesis that a random large enough and dense enough bipartite directed sub-graph of the web almost surely has a core. Note that a community may have multiple cores, a fact that emerged in experiments executed in [9]. Note also that the cores are directed: there is a set of  $i$  pages all of which hyperlink to a set of  $j$  pages. The  $i$  pages that contain the links are referred to as Fans and the  $j$  pages are referenced as centers.

Due to the large amount of data used to enumerate the cores, trawling includes several preliminary iterative and simple algorithms to minimize the data. First, HITS is applied to extract the non nepotistic Fans. Non nepotistic links means links to pages on other sites. Keeping non nepotistic Fans require keeping the Centers or authorities pointed to by these Fans or hubs. Second trawling eliminates existing communities that are mirrored repeatedly both in their fans and centers. To do this, a shingling method created by [5] that identifies and eliminates such duplicates, is adopted. Third, to delete pages that have large in-degree trawling depends on the following empirical fact :

The probability that a page has in-degree  $i$  is roughly  $1/i^2$  on one condition that  $i \leq 410$ .

Therefore, unusually popular pages such as [www.yahoo.com](http://www.yahoo.com) are eliminated



Fourth, trawling prunes centers by in-degree. It deletes all pages that are very highly referenced on the web such as the home pages of web portals. Therefore potential centers that have an in-degree greater than a carefully-chosen threshold  $K$  are pruned.

After these preliminary steps, while trawling is generating cores, it prunes all unnecessary pages. For instance, when looking for  $(i, j)$  cores, any potential fan with out-degree smaller than  $j$  can be pruned and the associated edges deleted from the graph. Similarly, any potential center with in-degree smaller than  $i$  can be pruned and the corresponding edges deleted from the graph; this process can be done iteratively: when a fan gets pruned, then some of the centers that it points to may have their in-degree fall below the threshold  $i$  and qualify for pruning as a consequence; similarly for centers. The next pruning strategy is called "inclusion-exclusion pruning".

At every step, it either eliminates a page from contention or outputs an  $(i, j)$  core.

To generate  $(i, j)$  core., trawling uses the following fact:

Let  $\{c_1, c_2, \dots, c_j\}$  be the centers adjacent to a fan  $x$ . Let  $N(c_t)$  denote the neighborhood of  $c_t$ , the set of fans that point to  $c_t$ . Then,  $x$  is a part of a core if and only if the intersection of the sets  $N(c_t)$  has size at least  $i$ .

In detail, trawling maintains a set  $S(x)$  corresponding to each fan  $x$ . The goal is that at the end of the computation, the set corresponding to the fan  $x$  will be the intersection of the sets  $N(c_t)$  specified in the above fact:

Stream through the edges sorted by centers. For each fan  $x$  adjacent to  $y$ ,  $x$  has an out-degree  $j$ , find the centers adjacent to  $x$ . Then execute the intersection of the corresponding  $N(c_t)$  such that

$$S(x) = N(c_1) \cap N(c_2) \dots \dots \dots \cap N(c_j)$$

If size  $(S(x) = i)$ , then an  $(i, j)$  core is generated

It is worth mentioning, that the running time of the above pruning steps is linear in the size of the input plus the number of communities produced in these steps.

Trawling offers many advantages. First, it can give an idea about fossilized communities. A fossil is a community core, not all of whose fan pages exist on the web

today. Second, recoverability for communities that have not fossilized: recovering today's community from the core extracted before, can be done by applying "Clever" on the fans.

## Chapter 3

### Implementation of H.I.T.S

#### 3.1 Introduction

What follows is a description of H.I.T.S implementation. As described earlier H.I.T.S passes through two major steps. The collection of data, and the implementation of the iterative algorithm that calculates and outputs the hub and authority weights for each page. This implementation is based on two query strings: "Lebanese tourism" and tourism in Lebanon. A detailed description of the data collection and implementation of the algorithm will be included. Furthermore a comparison between these two queries would be beneficial, especially that they refer to the same topic.

#### 3.2 Data collection

In H.I.T.S data collection is referred to as the formation or construction of a focused sub graph by the formation of the root set  $R_\sigma$  and the base set  $S_\sigma$ .  $R_\sigma$  will include the top twenty pages returned by applying the two query strings mentioned earlier to a web portal, in this case it is the search engine Altavista.  $S_\sigma$  is obtained by growing  $R_\sigma$  to include any page pointed to by a page in  $R_\sigma$  and any page that points to a page in  $R_\sigma$ . Practically we will be using the search engine Google in the construction of  $S_\sigma$ .

##### 3.2.1 The search engine Google

Google is a search engine that makes use of link structure [6] and anchor text to provide information for making relevance judgments and quality filtering. As the collection size of search engines' result is growing, a tool that has very high precision is needed. The number of relevant documents returned, say in the top tens of result, is a good start for a user who is unwilling to spend much of his time for searching between the results of an initial search. Beyond that, Google offers many features: it can display the pages similar to the site in query, and especially it can display all the pages that link to a specific site or URL. The latter advantage was one of the main reasons for choosing Google in the construction of the base set.

### 3.2.2 Construction of the root set $R_0$

As mentioned earlier,  $R_0$  will include the first twenty pages resulted from the query applied on the search engine Altavista.

For the first one , "lebanese tourism" the first twenty pages are the following:

- 1 [www.discoveredmonton.com/Edmonton/Restaurants/Lebanese](http://www.discoveredmonton.com/Edmonton/Restaurants/Lebanese)
- 2 [www.detroit.worldweb.com/Restaurants/Lebanese](http://www.detroit.worldweb.com/Restaurants/Lebanese)
- 3 [www.detroit.worldweb.com/Dearborn/Restaurants/Lebanese/](http://www.detroit.worldweb.com/Dearborn/Restaurants/Lebanese/)
- 4 [www.alj.com/companies/hotel/saudico.htm](http://www.alj.com/companies/hotel/saudico.htm)
- 5 [www.ottawa-hull.worldweb.com/Restaurants/Lebanese](http://www.ottawa-hull.worldweb.com/Restaurants/Lebanese) 4
- 6 [www.vancouver.worldweb.com/Restaurants/lebanese](http://www.vancouver.worldweb.com/Restaurants/lebanese)
- 7 [www.lebanon-tourism.gov.lb](http://www.lebanon-tourism.gov.lb)
- 8 [www.arabicnews.com/ansub/Daily/Day/990513/1999051344.html](http://www.arabicnews.com/ansub/Daily/Day/990513/1999051344.html)
- 9 [www.arabicnews.com/ansub/Daily/Day/990128/1999012839.html](http://www.arabicnews.com/ansub/Daily/Day/990128/1999012839.html)
- 10 [www.vancouver.worldweb.com/Vancouver/Restaurants/Lebanese](http://www.vancouver.worldweb.com/Vancouver/Restaurants/Lebanese)
- 11 [www.berro.com/Lebanon.htm](http://www.berro.com/Lebanon.htm)
- 12 [www.arabinfoseek.com/lebanon-travel\\_&\\_tourism.htm](http://www.arabinfoseek.com/lebanon-travel_&_tourism.htm)
- 13 [inic.utexas.edu/menic/countries/lebanon.html](http://inic.utexas.edu/menic/countries/lebanon.html)
- 14 [www.ventnouveau.com.lb/tourism.htm](http://www.ventnouveau.com.lb/tourism.htm)
- 15 [www.ventnouveau.com.lb](http://www.ventnouveau.com.lb)
- 16 [www.lebcom.com/mysearch/mysearch.cgi?category=Travel](http://www.lebcom.com/mysearch/mysearch.cgi?category=Travel)
- 17 [208.2.80.22/business/07\\_04\\_01\\_b.htm](http://208.2.80.22/business/07_04_01_b.htm)
- 18 [www.wlo-usa.org/Opinion/Ziad/Ziad9.htm](http://www.wlo-usa.org/Opinion/Ziad/Ziad9.htm)
- 19 [ils.unc.edu/~mehol/AANA/links.html](http://ils.unc.edu/~mehol/AANA/links.html)
- 20 [www.acropolis.com.lb/soffers.html](http://www.acropolis.com.lb/soffers.html)

For the second query, tourism in Lebanon, the first twenty pages output by the search engine Altavista are:

- 1 [www.lebanon.com](http://www.lebanon.com)
- 2 [www.lebanon.com/tourism](http://www.lebanon.com/tourism)
- 3 [www.lebanon-tourism.gov.lb](http://www.lebanon-tourism.gov.lb)
- 4 [www.1stlebanon.net](http://www.1stlebanon.net)
- 5 [www.lebanon-pages.com](http://www.lebanon-pages.com)
- 6 [www.lebindex.com](http://www.lebindex.com)
- 7 [www.libanmall.com](http://www.libanmall.com)
- 8 [www.llion.org](http://www.llion.org)
- 9 [www.tripoli-lebanon.com](http://www.tripoli-lebanon.com)
- 10 [www.lebanon-express.com](http://www.lebanon-express.com)
- 11 [www.oingo.com/topic/47/47574.html](http://www.oingo.com/topic/47/47574.html)
- 12 [www.tourismlebanon.com](http://www.tourismlebanon.com)
- 13 [www.cdl.com.lb](http://www.cdl.com.lb)
- 14 [www.bso.com.lb](http://www.bso.com.lb)
- 15 [www.saadtours.com](http://www.saadtours.com)
- 16 [www.printania.com](http://www.printania.com)
- 17 [www.lebanonlinks.com/le/touri.html](http://www.lebanonlinks.com/le/touri.html)
- 18 [www.embofleb.org/lebanon.htm](http://www.embofleb.org/lebanon.htm)
- 19 [www.swi-news.com/SWI-Lebanon.htm](http://www.swi-news.com/SWI-Lebanon.htm)
- 20 [www.arab-business.net](http://www.arab-business.net)

The choice of the two queries had several reasons. First the topic has a relevant meaning as it is about the tourism in Lebanon. Second the two queries were aimed at the same topic but each in different perspective. For the first one, even though it is about the tourism in Lebanon, but it was deliberately chosen as a string to narrow down as much as possible the search so it can be compared with the other sentence query which can support a wider variety of results. In fact the query string "Lebanese tourism" has resulted , when applied to the search engine , in all the sum of about 70 results. Meanwhile the sentence query, tourism in Lebanon, has resulted in all about 6000000 results. This choice has been done to analyze the behavior of the H.I.T.S algorithm towards the scarcity and the abundance problem.

### 3.2.3 Construction of the base set $S_\sigma$

As mentioned in section 2.1, Google offers the service of supplying the in links of any URL supplied. But unfortunately, due to the way Google crawls the web, it does not supply all the in links for a specified URL. Therefore, in the construction of the base set, first we get the in links and out links of each of the twenty pages mentioned above. For the in links we use Google. For the out links we use the source code of each page. But this will leave the base set probably with pages not connected to each other even though, in reality, they are. So to be sure that the base set is representing the reality of the query in the web, finding the out links of all the links already gathered until then is a must. With the exception of the pages within the same site. In this second step, the borders of the base set are widened more than what is required in H.I.T.S. But this will not affect the overall result; on the contrary it will be a good environment for the H.I.T.S to operate. For the first query, the base set  $S_\sigma$  has a size of 489 pages and 573 links. For the second query, the size was 704 pages and 989 links. We should mention that the size for both is after the elimination of duplicates that can arise due to the manual extraction of the links.

### 3.2.4 Data structure

The collection of data is then converted into a table, the links table. This table includes two fields: Master and Child. The Master field includes all pages that are referencing other pages. The Child field includes all pages that are being referenced by other pages. This table has two indices: Bymaster and Bychild. Another table is needed, the Sites table. This table includes five fields: a Site field that includes all the pages, Pointedbyx and Pointstoy fields are used to store the x and y weights for each page, Countx and County fields are used to store the number of in links and out links respectively. The latter two fields are used for the implementation of trawling later. The sites table has three indices: Bysite, used for trawling, Byx and Byy indices are used to output the results of H.I.T.S in descending order.

The two tables are filled in the following way: first the Links table is filled then for each record in the Links table sorted using Bymaster index, if the page is found in the sites table then the County field is incremented by one else a page is added and County is incremented by one. Then the Links table is sorted using Bychild index, and similarly for

each record if the page is found in the sites table then Countx field is added by one else the page is added and countx is incremented by one. At the end for each page in the Sites table, Pointedbyx and Pointstoy fields are initialized to one.

### 3.2.5 Computing hubs and authorities

After the construction of the base set, the H.I.T.S iterative algorithm is applied. Note that an explanation of this algorithm can be found in chapter 2 section 2.3.1

The number of iterations K executed by H.I.T.S can be manipulated. This implementation will show the variation of hubs and authorities priority according to their calculated weights as the threshold K is changing.

K=1		Query= "lebanese tourism"		Query= tourism in lebanon	
Top 10 authorities		Weight (x,y)	Top 10 authorities		Weight (x,y)
www.berro.com/lebanon.htm		(0.61,0.11)	www.saadtours.com		(0.21,0.00)
www.lebanon-tourism.gov.lb		(0.40,0.01)	www.llion.org		(0.18,0.01)
www.ventnouveau.com.lb		(0.23,0.03)	www.lebanon-express.com		(0.18,0.00)
www.worldweb.com		(0.21,0.08)	www.embofleb.org		(0.18,0.00)
www.ventnouveau.com.lb/tourism.htm		(0.17,0.04)	www.aub.edu.lb		(0.18,0.00)
canada.worldweb.com		(0.15,0.00)	www.arab-business.net		(0.18,0.01)
www.ontario.worldweb.com		(0.11,0.00)	www.mea.com.lb		(0.16,0.00)
www.berro.com		(0.11,0.00)	www.lebanon-tourism.gov.lb		(0.16,0.00)
www.annahar.com.lb		(0.11,0.00)	www.bdl.gov.lb		(0.16,0.00)
www.ottawa-hull.worldweb.com		(0.08,0.19)	www.printania.com		(0.14,0.00)

Table1 authority comparision for k = 1

K=1	Query= "lebanese tourism"	Query= tourism in lebanon	
Top 10 hubs	Weight (x,y)	Top 10 hubs	Weight (x,y)
<a href="http://inic.utexas.edu/menic/countries/lebanon.html">inic.utexas.edu/menic/countries/lebanon.html</a>	(0.00,0.98)	<a href="http://www.lebanonembassy.org/links/lebenon.html">www.lebanonembassy.org/links/lebenon.html</a>	(0.00,0.49)
<a href="http://www.keele.ac.uk/depts/por/mebase.htm">www.keele.ac.uk/depts/por/mebase.htm</a>	(0.00,0.10)	<a href="http://www.embofleb.org/lebanon.htm">www.embofleb.org/lebanon.htm</a>	(0.11,0.32)
<a href="http://www.embofleb.org/lebanon.htm">www.embofleb.org/lebanon.htm</a>	(0.00,0.08)	<a href="http://www.middleeastnews.com/lebaneselinks.html">www.middleeastnews.com/lebaneselinks.html</a>	(0.00,0.27)
<a href="http://ain-ebel.org/links.htm">ain-ebel.org/links.htm</a>	(0.00,0.08)	<a href="http://www.nala.com/links.htm">www.nala.com/links.htm</a>	(0.00,0.25)
<a href="http://www.usaidlebanon.org/lb/html/links1.htm">www.usaidlebanon.org/lb/html/links1.htm</a>	(0.00,0.05)	<a href="http://www.ain-ebel.org/links.htm">www.ain-ebel.org/links.htm</a>	(0.00,0.24)
<a href="http://www.lebanonembassy.org/links/lebenon.html">www.lebanonembassy.org/links/lebenon.html</a>	(0.00,0.05)	<a href="http://www.mountlebanon.org/links.html">www.mountlebanon.org/links.html</a>	(0.02,0.22)
<a href="http://ils.unc.edu/~mehol/aama/links.html">ils.unc.edu/~mehol/aama/links.html</a>	(0.00,0.03)	<a href="http://inic.utexas.edu/menic/countries/lebanon.html">inic.utexas.edu/menic/countries/lebanon.html</a>	(0.00,0.22)
<a href="http://almashriq.hiof.no/base/travel.html">almashriq.hiof.no/base/travel.html</a>	(0.01,0.02)	<a href="http://hani.ourfamily.com/lebanon.htm">hani.ourfamily.com/lebanon.htm</a>	(0.00,0.17)
<a href="http://www.tourist-office.org">www.tourist-office.org</a>	(0.00,0.02)	<a href="http://www.hri.org/nodes/mideast.html">www.hri.org/nodes/mideast.html</a>	(0.00,0.17)
<a href="http://www.syriatourism.org">www.syriatourism.org</a>	(0.00,0.02)	<a href="http://www.gksoft.com/govt/en/lb.html">www.gksoft.com/govt/en/lb.html</a>	(0.04,0.16)

Table2: hub comparison for k = 1

K=5	Query= "lebanese tourism"	Query= tourism in lebanon	
Top 10 authorities	Weight (x,y)	Top 10 authorities	Weight (x,y)
<a href="http://www.lebanon-tourism.gov.lb">www.lebanon-tourism.gov.lb</a>	(0.15,0.00)	<a href="http://www.aub.edu.lb">www.aub.edu.lb</a>	(0.24,0.00)
<a href="http://www.annahar.com.lb">www.annahar.com.lb</a>	(0.13,0.00)	<a href="http://www.bdl.gov.lb">www.bdl.gov.lb</a>	(0.22,0.00)
<a href="http://www.lp.gov.lb">www.lp.gov.lb</a>	(0.12,0.00)	<a href="http://www.lau.edu.lb">www.lau.edu.lb</a>	(0.20,0.00)
<a href="http://www.csb.gov.lb">www.csb.gov.lb</a>	(0.11,0.00)	<a href="http://www.lebanonart.com">www.lebanonart.com</a>	(0.17,0.00)
<a href="http://www.cib.gov.lb">www.cib.gov.lb</a>	(0.11,0.00)	<a href="http://www.ndu.edu.lb">www.ndu.edu.lb</a>	(0.17,0.00)
<a href="http://www.lau.edu.lb">www.lau.edu.lb</a>	(0.11,0.00)	<a href="http://www.bau.edu.lb">www.bau.edu.lb</a>	(0.17,0.00)
<a href="http://www.lebanon.com">www.lebanon.com</a>	(0.11,0.00)	<a href="http://www.mtv.com.lb">www.mtv.com.lb</a>	(0.16,0.00)
<a href="http://www.mmorning.com">www.mmorning.com</a>	(0.11,0.00)	<a href="http://www.mea.com.lb">www.mea.com.lb</a>	(0.16,0.00)
<a href="http://www.lebanese-forces.org">www.lebanese-forces.org</a>	(0.11,0.00)	<a href="http://www.middleeastnews.com">www.middleeastnews.com</a>	(0.14,0.02)
<a href="http://www.ahrar.org.lb">www.ahrar.org.lb</a>	(0.11,0.00)	<a href="http://www.lebanonlinks.com">www.lebanonlinks.com</a>	(0.14,0.11)

Table3: authority comparison for k= 5



K=5	Query= "lebanese tourism"	Query= tourism in lebanon	
Top 10 hubs	Weight (x,y)	Top 10 hubs	Weight (x,y)
inic.utexas.edu/menic/countries/lebanon.html	(0.00,0.98)	www.lebanonembassy.org/links/lebenon.html	(0.00,0.71)
www.keele.ac.uk/depts/por/mebase.htm	(0.00,0.11)	www.embofleb.org/lebanon.htm	(0.01,0.40)
www.embofleb.org/lebanon.htm	(0.00,0.08)	www.ain-ebel.org/links.htm	(0.00,0.26)
ain-ebel.org/links.htm	(0.00,0.06)	www.middleeastnews.com/lebaneselinks.html	(0.00,0.26)
www.usaidlebanon.org.lb/html/links1.htm	(0.00,0.05)	www.nala.com/links.htm	(0.00,0.21)
www.lebanonembassy.org/links/lebenon.html	(0.00,0.03)	inic.utexas.edu/menic/countries/lebanon.html	(0.00,0.19)
ils.unc.edu/~mehol/aana/links.html	(0.00,0.02)	www.mountlebanon.org/links.html	(0.00,0.15)
www.ottawa-hull.worldweb.com	(0.01,0.02)	www.lebanonlinks.com	(0.14,0.11)
almashriq.hiof.no/base/travel.html	(0.00,0.01)	www.gksoft.com/govt/en/lb.html	(0.01,0.11)
www.tourist-office.org	(0.00,0.01)	www.hri.org/nodes/mideast.html	(0.00,0.10)

Table4: hub comparison for k=5

K=9	Query= "lebanese tourism"	Query= tourism in lebanon	
Top 10 authorities	Weight (x,y)	Top 10 authorities	Weight (x,y)
www.lebanon-tourism.gov.lb	(0.14,0.00)	www.aub.edu.lb	(0.24,0.00)
www.annahar.com.lb	(0.13,0.00)	www.bdl.gov.lb	(0.22,0.00)
www.lp.gov.lb	(0.12,0.00)	www.lau.edu.lb	(0.20,0.00)
www.csb.gov.lb	(0.11,0.00)	www.ndu.edu.lb	(0.17,0.00)
www.cib.gov.lb	(0.11,0.00)	www.lebanonart.com	(0.17,0.00)
www.lebanon.com	(0.11,0.00)	www.bau.edu.lb	(0.17,0.00)
www.morning.com	(0.11,0.00)	www.mtv.com.lb	(0.16,0.00)
www.lebanese-forces.org	(0.11,0.00)	www.mea.com.lb	(0.16,0.00)
www.ahrar.org.lb	(0.11,0.00)	www.middleeastnews.com	(0.14,0.02)
assa.fir.com	(0.11,0.00)	www.lebanonlinks.com	(0.14,0.11)

Table5: authority comparison for k = 9

K=9	Query= "lebanese tourism"	Query= tourism in lebanon	
Top 10 hubs	Weight (x,y)	Top 10 hubs	Weight (x,y)
<a href="http://inic.utexas.edu/menic/countries/lebanon.html">inic.utexas.edu/menic/countries/lebanon.html</a>	(0.00,0.99)	<a href="http://www.lebanonembassy.org/links/lebenon.html">www.lebanonembassy.org/links/lebenon.html</a>	(0.00,0.71)
<a href="http://www.keele.ac.uk/depts/por/mebase.htm">www.keele.ac.uk/depts/por/mebase.htm</a>	(0.00,0.10)	<a href="http://www.embofleb.org/lebanon.htm">www.embofleb.org/lebanon.htm</a>	(0.01,0.40)
<a href="http://www.embofleb.org/lebanon.htm">www.embofleb.org/lebanon.htm</a>	(0.00,0.08)	<a href="http://www.ain-ebel.org/links.htm">www.ain-ebel.org/links.htm</a>	(0.00,0.26)
<a href="http://ain-ebel.org/links.htm">ain-ebel.org/links.htm</a>	(0.00,0.06)	<a href="http://www.middleeastnews.com/lebaneselinks.html">www.middleeastnews.com/lebaneselinks.html</a>	(0.00,0.26)
<a href="http://www.usaidlebanon.org.lb/html/links1.htm">www.usaidlebanon.org.lb/html/links1.htm</a>	(0.00,0.05)	<a href="http://www.nala.com/links.htm">www.nala.com/links.htm</a>	(0.00,0.20)
<a href="http://www.lebanonembassy.org/links/lebenon.html">www.lebanonembassy.org/links/lebenon.html</a>	(0.00,0.03)	<a href="http://inic.utexas.edu/menic/countries/lebanon.html">inic.utexas.edu/menic/countries/lebanon.html</a>	(0.00,0.19)
<a href="http://ils.unc.edu/~mehol/aana/links.html">ils.unc.edu/~mehol/aana/links.html</a>	(0.00,0.02)	<a href="http://www.mountlebanon.org/links.html">www.mountlebanon.org/links.html</a>	(0.00,0.15)
<a href="http://aimashriq.hiof.no/base/travel.html">aimashriq.hiof.no/base/travel.html</a>	(0.00,0.01)	<a href="http://www.lebanonlinks.com">www.lebanonlinks.com</a>	(0.14,0.11)
<a href="http://www.tourist-office.org">www.tourist-office.org</a>	(0.00,0.01)	<a href="http://www.gksoft.com/govt/en/lb.html">www.gksoft.com/govt/en/lb.html</a>	(0.01,0.11)
<a href="http://www.syriatourism.org">www.syriatourism.org</a>	(0.00,0.01)	<a href="http://www.wahat.com/home.html">www.wahat.com/home.html</a>	(0.00,0.10)

Table6: hub comparison for k =9

A quick comparison for the two queries reveals that the top authorities for both queries did not converge totally into sites that are directly related and relevant to the tourism in Lebanon. While only two pages relevant to the main topic were considered as authorities in the query string "Lebanese tourism", the other eight ones were less relevant. They are related to either the Lebanese media or to governmental institutions. On the other hand, we find four of the top authorities for the second query, strongly relevant to the education in Lebanon. Two sites are relevant to the media, and three sites are directly relevant to the topic in query.

From the point of view hubs, four of the top results for the string query generalized to include sites about tourism either on the world or other countries. Meanwhile most of the top hubs for the tourism in Lebanon query were strongly connected and relevant to the tourism in Lebanon.

This is a problem that H.I.T.S can fall into it. If some pages linking to the root set are highly referenced but are more or less irrelevant to the topic they can be output as top authorities. Eventhough the results for the query sentence tourism in Lebanon were more representative then the first one but H.I.T.S did not rank the relevant pages to tourism in

the first five authorities. That is why including additional search techniques like anchor text to the H.I.T.S might be more beneficial.

Note that more experiments could have been executed if an archive of the World Wide Web was available or at least a sample of it, especially that the data input used is extracted manually and it could be subject to human error.

After the implementation of H.I.T.S, the following chapter will discuss the implementation of "Trawling the web for emerging cyber communities".

## Chapter 4

### Implementation of Trawling

#### 4.1 Introduction

This chapter will include a description of how Trawling is implemented. This implementation will be based on the query sentence tourism in Lebanon. The choice fall on this query due to the amount of links that is found in the base set which will be the data input used. I will include a description of the Trawling's algorithm, some output examples and the number of cores generated.

#### 4.2 Optimization

Trawling passes through several preliminary steps for optimization of the algorithm. These steps are useful for a large amount of data in the order of millions of pages. These steps are simple elimination procedures such as pruning centers by in degree, eliminating popular pages that have large pages pointing to it, such as web portal pages, or eliminating mirrors by using shingling methods. In our case, due to the way the data was collected, these reduction techniques are omitted. The inclusion exclusion- technique, described in chapter 2 section 2.3.2, will be preserved. This is because the inclusion-exclusion technique is executed simultanesouly with the core generation.

#### 4.3 Trawling

The data source used is the one relative to the query tourism in Lebanon. But I added all the out links including those which are nepotistic ones This is done because I wanted to enumerate all possible bipartite cores. The data source's volume increased to include 806 pages with 1520 links in all. The same data structure as the one in H.I.T.S is used. This is due to the fields I already added. The pages in the master field will be called Fans and the ones of the child field as Centers.

The basic concept of the generation of cores  $c(i, j)$  is the following:

Sort fans of outdegree  $j$ .

Sort centers of indegree  $i$ .

For each fan of outdegree  $j$  do

Find  $S(F)$ , the set of centers pointed to by  $F$

For each center  $C \in S(F)$  do

Find  $N(C)$  set of fans pointing to  $C$

If size  $\{ \cap N(C) \} = i$  then output the core  $C(i, j)$ .

each fan having a number of links equal to  $j$  will be kept in another (master, child) table. The Links table that is being used is static and it will not be modified, but a dynamic virtual copy will be used to eliminate either fans or centers according to their criterion. The same process will be used to sort centers. Note that the language I am using for implementation supports SQL queries, but I preferred to use this way so it can be applied to any programming language.

The running time of the above algorithm is the following:

for sorting fans with out degree  $j$ :  $O(\text{number of records in the links table})$ .

same for sorting centers with indegree  $i$ .

for the generation of  $c(i, j)$  cores:  $O(\{\text{number of fans of outdegree } j\}^i * j)$

From the following results, there is 87  $c(1, j)$  cores out of 124 cores in total. These cores, even though they are bipartite cores by concept, but in reality they are directed stars, where one fan is pointing to  $j$  centers; but this is not all, these  $j$  centers are not referred to by any other fans at the same time. On the other hand, there is 31 out of 124 cores of the form  $c(i, 1)$ . These cores represent directed star structure too, where  $i$  fans are pointing to one center and there is no other center that these fans point to at the same time. For the bipartite cores  $c(i, j)$  where  $i$  and  $j$  are different than one, we find 6 out of 124 different ones. Out of these six cores there were only one non nepotistic core.

The following table shows the resulting cores:

i	j	# of cores	i	j	# of cores	i	j	# of cores
1	2	17	1	16	1	1	37	1
1	3	6	1	17	3	1	38	1
1	4	6	1	18	1	2	1	5
1	5	8	1	19	1	2	2	2
1	6	4	1	20	1	2	3	3
1	7	5	1	21	1	3	1	1
1	8	2	1	22	1	4	1	4
1	9	4	1	24	2	4	2	1
1	10	4	1	26	1	5	1	7
1	11	4	1	27	1	6	1	3
1	12	4	1	29	2	7	1	11
1	13	3	1	35	1			
1	15	1	1	36	1			

Table 7: bipartite core list

As an example is the following three cores:

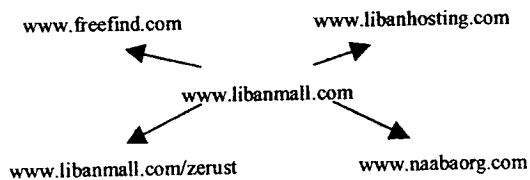


fig.3 core c(1,4)

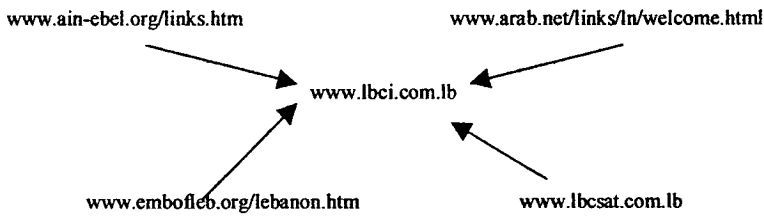


fig 4 core  $\alpha(4,1)$



fig 5 core  $\alpha(2,3)$

In this chapter, cores were generated. There can be different structures of cores; even though they all are bipartite cores, but some of them can be called Stars due to their structure. There is two kinds of stars, the in going star and the out going star, but all are directed. We found six bipartite core where  $i$  and  $j$  are different then one. Out of these six, one was non-nepotistic. This shows that the other cores may be an artificially established community serving commercial purposes, rather than a spontaneously emerging web community.

The following chapter discusses the formation of other structures on the web: the tree structure .

## Chapter 5

### Tree structure

#### 5.1 Introduction

In 1994, one of the first web search engines, the World Wide Web Worm (WWWW) had an index of 110,000 web pages and web accessible documents. As of November 1997, the top search engines claim to index from 2 million ( Webcrawler ) to 100 million web documents ( from Search Engine Watch) [3]. It is foreseeable by this year, 2001, a comprehensive index of the web will contain over a billion documents. At the same time, the number of queries search engines handle has grown incredibly too. In march and April 1994, the World Wide Web Worm received an average of about 1500 queries per day. In November 1997 Altavista claimed it handled roughly 20 million queries per day. With the increasing number of users on the web, the expansion of the world wide web, and automated systems which query search engines, it is likely, that top search engines, will handle hundreds of millions of queries per day by the year 2001. Our goal is to try to find specific structures on the web that might help reducing the problems caused by the growth and increase of the World Wide Web.

#### 5.2 Tree structure

The idea is based on Kleinberg's mutual reinforcement relationship between web pages [11] and the co-citation concept [9]. Kleinberg elaborated an algorithm, the H.I.T.S algorithm, which translates the mutual reinforcement relationship into hub and authority weights for each page. In [9] trawling enumerates all the bipartite cores found using an archive of the internet.

Trying to find tree structure based on bipartite cores can be helpful in many ways. First it can be an early step for structuring and classifying topics on the World Wide Web. Second it can help reduce the search in communities where bipartite cores and tree



structures can be their nuclei. Third, the structure of the tree can help search engines in indexing, by including a parser that can parse tree structures starting from their nodes.

Consider a tree based structure, starting from the root, one can reach the leaf nodes using a parsing methodology. As an example is the binary tree:

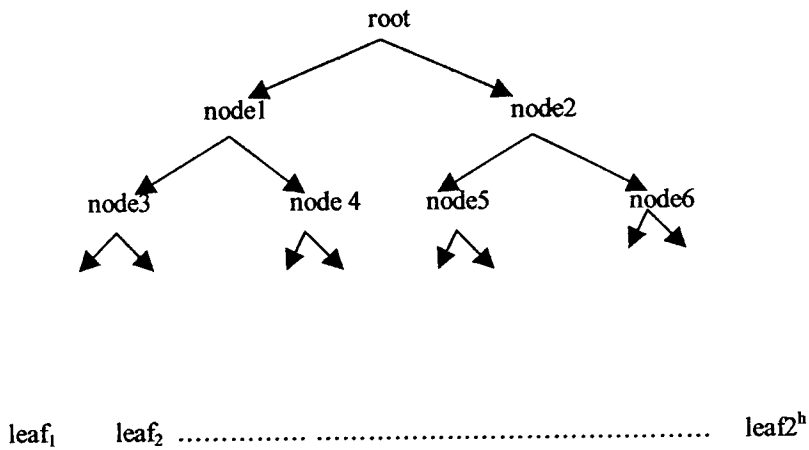


fig. 6: binary tree

In the binary tree of height  $h$  there is  $2^h$  leaves and in all there is  $2^{h+1} - 1$  nodes.

Consider now a bipartite core  $c(i, j)$  it has the following shape:

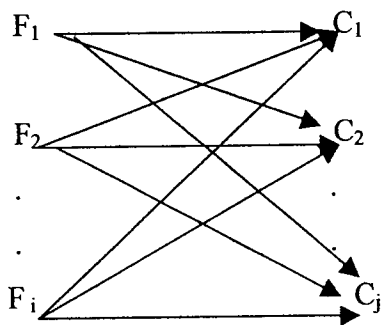


Fig. 7: core  $c(i, j)$

We will trawl the data input  $(i - 1)$  times on the fans obtained in the core to generate a tree of this form:

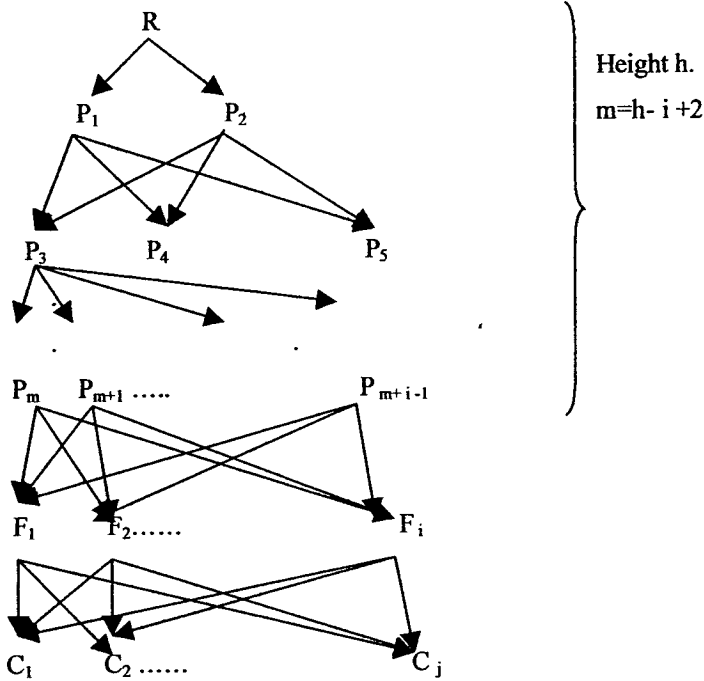


Fig.8: bipartite tree

Note that this tree supports  $i + j + \sum_{k=1 \dots (h+1)} k = (h+2) * (h+1)/2 + i + j = i * (i + 1)/2 + j$  nodes. Because  $h+1 = i - 1 \implies h = i - 2$ .

### 5.2.1 Tree generation

As I mentioned earlier, the generation of this tree shape is based on the bipartite cores:

For each fan in the core  $c(i, j)$

first find the set of pages  $N(F_i)$  from the links table that points to it.

If size  $\{N(F_1) \cap N(F_2) \dots \cap N(F_i)\} = i - 1$  then

$$S = \{N(F_1) \cap N(F_2) \dots \cap N(F_i)\}$$

$$i = i - 1$$

repeat the same process until  $i = 1$  or  $S$  is empty.

If  $i$  is equal to one then a tree shape is generated.

Note that this algorithm is the same used to trawl the web for communities. At the maximum it will iterate  $i - 1$  times. Note also that this procedure for tree generation can be applied only on cores having  $i$  and  $j$  different than one.

### 5.2.2 Implementation

The data input used is the one for enumerating and generating bipartite cores. As we mentioned earlier this process depends on the results output by trawling algorithm. The implementation in chapter 4 revealed the birth of six bipartite cores, where  $i$  and  $j$  are different from one. Out of these cores only one can form a tree. The corresponding core has a size  $C(2,2)$ . And the tree structure is the following:

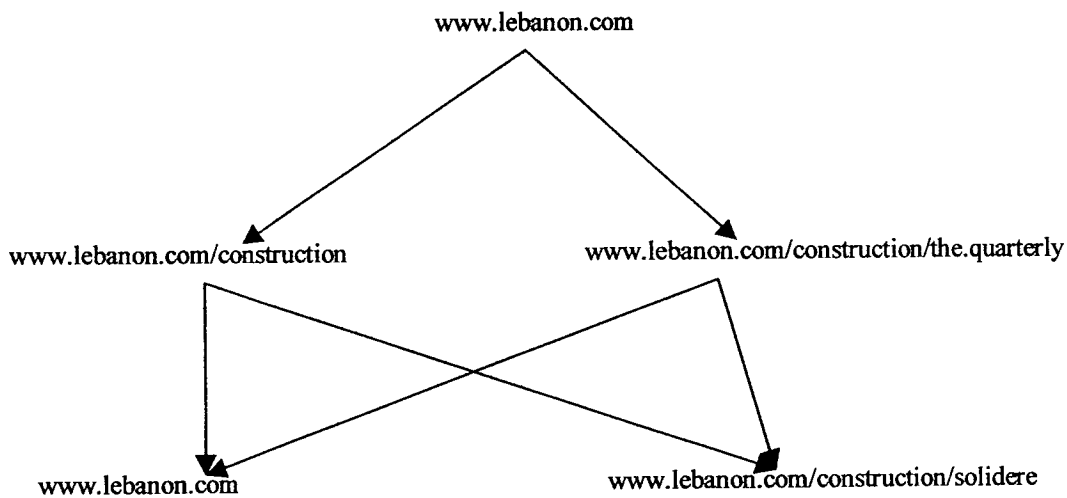


fig 9:resulting tree

As we can notice in this tree, even though it verifies the characteristics described earlier, one leaf node is the same as the root. This is due to the nepotistic core and the ring effect found between many pages on the web. Bipartite cores exhibit a tight relation between web page links, where pages of same topic tend to be referenced together. And this tree structure has even more tight bond than bipartite cores.

In this chapter the study of tree structure revealed their absence, at least tree structures of the form described above. Maybe other kind of trees having less tight bond exists, or maybe a more representative data source could help understanding the real existence of

tree structures. The following chapter will be dealing with webring structure, their existence, and how to identify them.

## Chapter 6

### ' Web Ring ' structure

#### 6.1 Introduction

The World Wide Web is a corpus that has been for years growing randomly and chaotically. Many releases and researches handled this subject. Kleinberg, in [11], gave birth to the famous H.I.T.S. In [9], the authors enumerated newly emerging communities by generating their bipartite cores. Can there be other forms of structures that can gather around it web pages to form communities?. More specifically can there be a significant number of ring structures or 'Web Rings' on the web?. There are several reasons for extracting such structures from the web. These 'Web Rings', if found, may provide a valuable resource; on the other hand studying them gives an insight on the sociology of the web and its evolution. We should not forget that 'Web Rings', by definition, are rings which end where they start. Therefore, with the increase in number of internet users and surfers, it would be an interesting idea if we can reduce the number of "hyperjumps" between pages by organizing the hyperlinks. More over 'Web Rings' might be the beginning solution for the abundance problem related to search engine's queries.

#### 6.2 'Web Rings'

By definition, as its name implies, a webring is a continuous, yet expanding group of linked web sites having common interests. If you were to start on a particular site and go through the entire sites on a webring, eventually you would end up back where you began.

For instance a ring of size four is of the form:

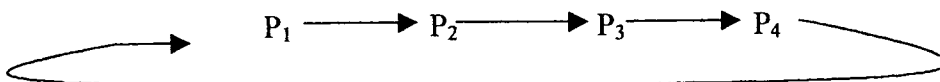


Fig.10: webring



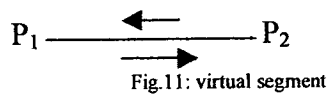
'Web Rings' can occur in at least three situations. First, if a topic includes several subtopics all are related, second when hyperlinks are added for surfing purposes such as "home page" hyperlink, third if sites sharing a common topic are referencing each other. In all ways, it is interesting to try enumerating these rings.

### 6.2.1 Advantages of 'Web Rings'

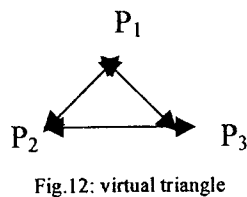
There are several theories talking, of advantages, 'Web Rings' can offer. For one, they can be an alternative for search engines. This is due to the virtual shape these 'Web Rings' can take. Another reason is, there is no "dead" sites on a webring. Most search engines update they're information upon notification, which means an outdated site can remain for a long time, and it can contribute in a query result no matter how outdated or incorrect that site or information is wrong. On the other hand, 'Web Rings' are small enough so that the contributing websites can be updated very often. Any website who is not valid anymore can be eliminated from the webring, thus saving the user of being frustrated by reaching a "dead" end site, which no longer exists. More over, contrary to kleinberg's theory, there are no favorites or hierarchies. In a webring, all sites are equal.

### 6.2.2 Virtual shape

Rings can be represented by virtual shapes. For instance a ring of size two can be represented as a segment:



A ring of size three can be represented as a virtual triangle:



A ring of size four can be represented as a virtual tetrad :

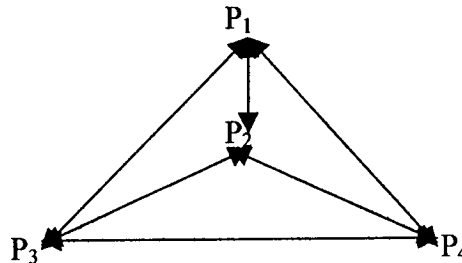


Fig.13: virtual tetrad

We can give many virtual shapes. Virtual means that physically, in any ring, not all nodes have direct links between each other as represented in the above figures. But due to the nature of the ring, where each node can reach another one directly or by passing through other nodes, we can elaborate virtual links between each two nodes. These virtual links can be represented by a list of hyperlinks on the main site. Therefore each webring can be transformed into a list of hyperlinks based on the main site. This list can be managed and updated easily. Especially that most 'Web Rings', are very topic specific. If these 'Web Rings' are converted as I mentioned, the relevant sites can be viewed randomly each by surfing one step only backward or forward. Therefore, instead of forcing a user to plow through thousands of search engine results to find a desired topic, converting related 'Web Rings' into a list of hyperlinks will enable the surfer to move between related sites just by one step. Note for instance, that a ring with  $n$  links, can be considered as a bi-directional star having  $n$  centers all pointing to each other.

### 6.2.3 Enumerating 'Web Rings'

In the course of designing an algorithm to enumerate rings on the web, several preliminary steps has to be executed:



Floyd & Warshall discovered a dynamic programming method to compute the shortest path between every pair of vertices in a graph of n nodes ( known as all-pairs-shortest-path). Consider the matrix

$$D = \begin{matrix} & U_1 & U_2 & U_3 & \dots\dots\dots & U_n \\ \begin{matrix} U_1 \\ U_2 \\ U_3 \\ \cdot \\ U_n \end{matrix} & \left( \begin{array}{cccccc} & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{array} \right) \end{matrix}$$

$$D^0[i,j] = \begin{cases} \text{Length of edge } (U_i, U_j) \text{ if it exists.} \\ \text{Infinity otherwise.} \end{cases}$$

The dynamic programming characterization is:

$$D^k [i ,j] = \text{minimum } ( D^{k-1} [i ,j] , D^{k-1} [i ,k] + D^{k-1} [k,j] )$$

Where  $D^k [i ,j]$  is the shortest path from i to j visiting only  $U_1$  to  $U_k$  along the way.

A simple variation of Floyd-Warshall can be developed to compute the reachability matrix R defined as follows

$$R[i,j] = \begin{cases} 1 & \text{if node j is reachable by node i.} \\ 0 & \text{otherwise.} \end{cases}$$

The dynamic programming characterization is:

$$R^k [i ,j] = (R^{k-1} [i ,j] \vee R^{k-1} [i ,k] \wedge R^{k-1} [k,j] )$$

### 6.2.3.4 Ring algorithm

The ring algorithm is based on the reachability matrix. Suppose that the number of pages resulting after the intersection described earlier is  $n$ . Then applying the dynamic programming characterization will result the following tree:

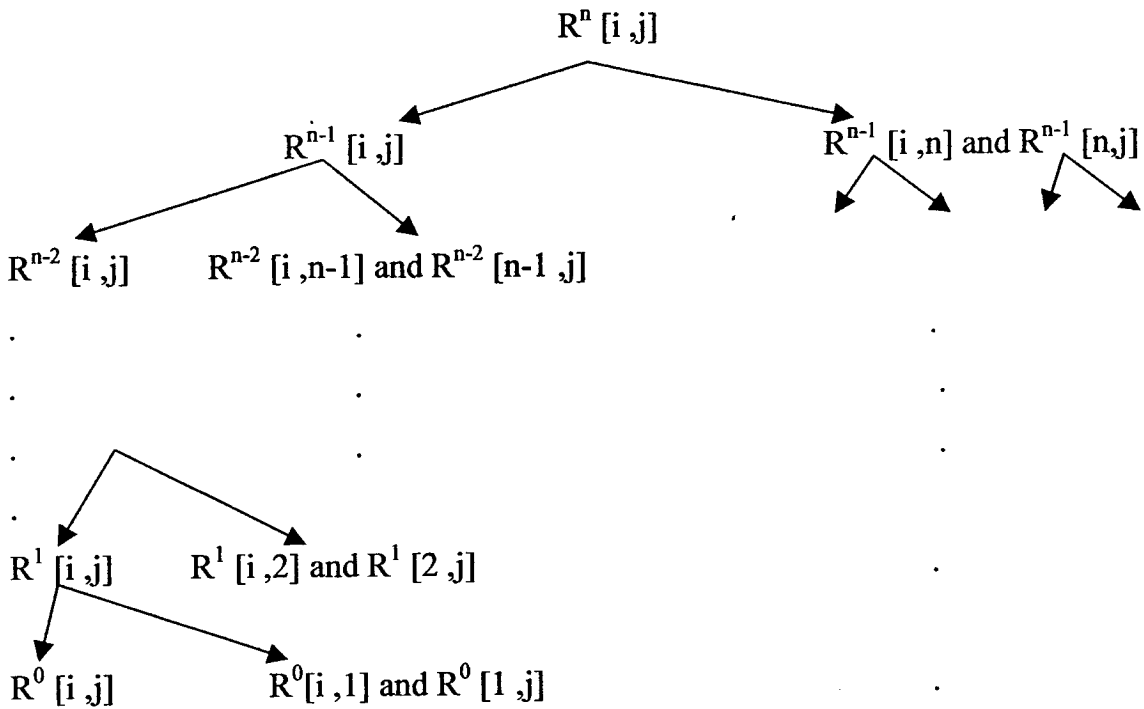


Fig 14: resulting ring tree

Based on this tree, the algorithm is the following:

```

For k = 1 to n
  For i = 1 to n
    For j = 1 to n
      If  $R[i, k] \neq 0$  and  $R[k, j] \neq 0$  then
         $R[i, j] = R[i, k] + R[k, j]$ 
  
```

The diagonal of the reachability matrix represent the rings .So after one iteration for K, we go through all the matrix checking if the value of  $R[ i ,i]$  is not zero, if yes then a ring is formed and  $R[i , i ]$  includes the size of the web ring. Every newly formed result is saved in an array representing the size of the ring.

Let F be the set of pages intersecting between the master field and the child field in the link table (table7) below:

$$F = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\}$$

Then the reachability matrix is initialized as follows:

$$R^0 = \begin{matrix} & \begin{matrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \end{matrix} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

After the execution of the program these are the following resulting rings:

$C_3 \rightarrow C_1 \rightarrow C_3$   
 $C_6 \rightarrow C_3 \rightarrow C_6$   
 $C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3$   
 $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_1$   
 $C_2 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2$   
 $C_6 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_6$   
 $C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3$   
 $C_6 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6$   
 $C_4 \rightarrow C_5 \rightarrow C_6 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4$   
 $C_5 \rightarrow C_6 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_5$   
 $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6 \rightarrow C_3 \rightarrow C_1$   
 $C_2 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2$   
 $C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3$   
 $C_6 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6 \rightarrow C_3 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6$

As we can notice there are some webrings which are formed by the same repeating webring , this is due to the fact that the newly emrging webrings are not extracted from the diagonal of the matrix and therefore they might contribute in the formation of other webrings. But the main advantage of such an algorithm is that it identifies all large webrings including those that are formed by nodes which are not being repeated or do not include recurring webrings but this happens on behalf of some small webrings which are not identified by this algorithm. Note that if the size of the reachability matrix is  $n \times n$ , then the running time algorithm is  $O(n^3)$ .

A heuristic that I applied on the reachability algorithm is the following:

I preserved all the links with which the reachability matrix was initialized and I added a test such that any entry that has been initialized was kept without modification. I tried to work only with entries that do not have links. So every time a new link is formed in any entry it will be preserved from further modifications. And any newly emerging webring is extracted from the webring , so it can not contribute in any other webring. This heuristic is represented by the following algorithm:

The reachability matrix is initialized in the following way:

$$R[i,j] = \begin{cases} U_i \longrightarrow U_j & \text{if } U_j \text{ is reachable by } U_i. \\ \text{Empty} & \text{otherwise.} \end{cases}$$

Then the reachability matrix is represented as follows:

$$R^0 = \begin{matrix} & \begin{matrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \end{matrix} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{matrix} & \left( \begin{array}{cccccccc} & & & & & & & \\ & C_1 \rightarrow C_2 & C_1 \rightarrow C_3 & C_1 \rightarrow C_4 & & & & \\ & & C_2 \rightarrow C_3 & C_2 \rightarrow C_4 & C_2 \rightarrow C_5 & & & \\ C_3 \rightarrow C_1 & & & C_3 \rightarrow C_4 & C_3 \rightarrow C_5 & C_3 \rightarrow C_6 & C_3 \rightarrow C_7 & \\ & & & & C_4 \rightarrow C_5 & C_4 \rightarrow C_6 & C_4 \rightarrow C_7 & C_4 \rightarrow C_8 \\ & & & & & C_5 \rightarrow C_6 & C_5 \rightarrow C_7 & C_5 \rightarrow C_8 \\ & & C_6 \rightarrow C_3 & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{array} \right) \end{matrix}$$

Based on the heuristic, the algorithm becomes as follows:

For k = 1 to n

    For i = 1 to n

        For j = 1 to n

            If R[i ,j] = empty then

                If R[i ,k] ≠ empty and R[k,j] ≠ empty then

                    R[i ,j] = U<sub>i</sub> → U<sub>k</sub> → U<sub>j</sub>

After the execution of the program these are the following resulting rings:

C3 → C1 → C3  
 C6 → C3 → C6  
 C3 → C1 → C2 → C3  
 C6 → C3 → C4 → C6  
 C6 → C3 → C5 → C6  
 C4 → C6 → C3 → C1 → C4  
 C5 → C6 → C3 → C4 → C5  
 C4 → C6 → C3 → C1 → C2 → C4  
 C5 → C6 → C3 → C1 → C2 → C5

Master	Child	Master	Child	Master	Child
C <sub>1</sub>	C <sub>2</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>9</sub>
C <sub>1</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>6</sub>	C <sub>7</sub>	C <sub>11</sub>
C <sub>1</sub>	C <sub>4</sub>	C <sub>4</sub>	C <sub>7</sub>	C <sub>7</sub>	C <sub>10</sub>
C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>8</sub>	C <sub>7</sub>	C <sub>12</sub>
C <sub>2</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>	C <sub>9</sub>
C <sub>2</sub>	C <sub>5</sub>	C <sub>5</sub>	C <sub>7</sub>	C <sub>8</sub>	C <sub>10</sub>
C <sub>3</sub>	C <sub>1</sub>	C <sub>5</sub>	C <sub>8</sub>	C <sub>8</sub>	C <sub>11</sub>
C <sub>3</sub>	C <sub>4</sub>	C <sub>6</sub>	C <sub>10</sub>	C <sub>8</sub>	C <sub>12</sub>
C <sub>3</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>11</sub>	C <sub>8</sub>	C <sub>9</sub>
C <sub>3</sub>	C <sub>6</sub>	C <sub>6</sub>	C <sub>12</sub>		
C <sub>3</sub>	C <sub>7</sub>	C <sub>6</sub>	C <sub>3</sub>		

Table8: links table

Note that if the size of the reachability matrix is  $n \times n$ , then the running time algorithm is  $O(\text{number of rings generated} * n^3)$ .

### 6.3 Implementation

The data source used for the implementation is the one based on the query sentence tourism in Lebanon. This data source was extracted from the web manually. It consumed three long days to be able to get a data set large enough especially for this implementation. As a result, after converting the data set into the links table and eliminating all duplicates, a total of 806 pages and 1520 links. Which is rather a decent representative sample of the WWW.

After optimization, a set of 76 pages was used in the reachability matrix. The 76 pages are those found in the master field and the child field. Therefore the size of the matrix is 76 x 76.

After running the program, based on the reachability matrix without applying any heuristics these were the top 100 webrings ranked according to their size:

Size	Number	Size	Number	Size	Number	Size	number
2	39	42	7	336	7	2687	2
3	12	43	3	337	3	2689	7
4	24	49	3	497	2	2690	3
5	8	62	1	510	1	4051	2
6	8	64	19	512	19	4064	1
7	3	82	2	513	1	4066	19
8	22	84	7	670	2	4067	1
10	8	85	3	672	7	8117	2
11	3	113	3	673	3	8130	1
12	1	126	1	1009	2	8132	19
13	2	128	19	1022	1	8133	1
14	2	129	1	1024	19	16249	2
16	20	166	2	1025	1	16262	1
17	3	168	7	1342	2	16264	19
19	2	169	3	1344	7	16265	1
21	7	241	2	1345	3	32513	2
22	3	254	1	2018	2	32526	1
30	1	256	19	2031	1	32528	19
32	19	257	1	2033	19	32529	1
40	2	334	2	2034	1	65041	2

There are more webrings which are sized in the order of millions. Of course these very large webrings are artificial ones because they are formed by other smaller recurring webrings.

On the other hand after applying the heuristics these were the following results:

107 rings of size 2.

53 rings of size 3.

18 rings of size 4.

2 rings of size 5.

Although these webrings are formed by some of the 76 nodes but obviously this is a partial result.

## **Conclusion**

From the above results it shows that web rings are more or less abundant. Web rings can play the substitute of search engines in order to eliminate the abundance problem. Furthermore they can reduce the surfing on the web. On the other hand, pages referencing other pages which are on the ring, can reference just the main page in the ring which can help reducing the number of hyperlinks.

In this chapter I tried to explain the advantages of the 'Web Rings', and their effect. I used a variant of the "all pairs shortest path " matrix to enumerate the 'Web Rings' and output the related results. But it did not give the desired results or at least it output all the possible links that can exist in a web ring even the existence of recurrent webrings within the same one. On the other hand I modified the reachability matrix by applying some heuristics on the algorithm , it outputted some results of small webrings but it did not identify any webrings of large sizes.



## Conclusion

In this research, I discussed the World Wide Web structure. I tried to offer a case study analysis of different structures available and I tried to find new web structures.

I applied the H.I.T.S algorithm on two queries discussing the same topic. Then I made comparison between the two results which revealed that Kleinberg's algorithm can be more operational if additional techniques are used. .

More over, based on some releases, I enumerated the bipartite cores found on a data source extracted manually from the web. Bipartite cores are the nuclei of communities, based on these bipartite cores , I tried to find tree structures on the web. These tree structures are tighter in bond between pages then bipartite cores. Which explains their absence.

On the other hand I used the reachability matrix to prove that web rings exist on the web, and they are abundant. One limitation to this work was the data input. Otherwise , there might have been rings of larger sizes, and 'Web Rings' between different sites

The World Wide Web is an interesting domain where out of anarchic links, structures are being formed. These structures can be helpful in classification of related topics into communities. Often, these communities are based on concentrated sub communities such as bipartite cores or 'Web Rings'. Classifying pages into communities can be helpful for web portals in their queries. Classification can be helpful for web portals in their queries.

## REFERENCES

- [1] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridar Rajagopalan, Raymie Stata, Andrew Tomkins, and Jant Wiener. Graph structure in the web. In *Proceedings of the Ninth International World Wide Web Conference*, 2000.
- [2] Albert, Jeong, and Barabasi 99. R. Albert, H. Jeong, and Barabasi. *Diameter of the world wide Web*, *Nature* 401:130-131, sep 1999.
- [3] Barabasi and Albert 99. A. Barabasi and R. Albert. *Emergence of scaling in random networks*, *Science*, 286(509), 1999.
- [4] Bharat and Henzinger 98. K. Bharat, and M. Henzinger . *Improved algorithms for topic distillation in hyperlinked environments*, Proc 21<sup>st</sup> SIGIR, 1998.
- [5] Broder et al. 97. Andrei Broder, Steve Glassman, Mark Manasse, and geoffrey Zweig. *Syntactic clustering of the web*. In proceedings of the 6<sup>th</sup> international World Wide Web conference, April 1997, pages 391-404.
- [6] Brin and Page 98. S. Brin, and L. Page. *The anatomy of a large scale hypertextual web search engine*, proc. 7<sup>th</sup> WWW, 1998.
- [7] Chakrabarti et al 98. S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. *Automatic resource compilation by analyzing hyperlink structure and associated text*, Proc. 7<sup>th</sup> WWW, 1998.
- [8] Kumar et al. 99. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. *Extracting large scale knowledge basis from the web*, proc. VLDB, jul 1999.
- [9] Kumar et al. 99. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. *Trawling the web for cyber communities*, Proc. 8<sup>th</sup> WWW, Apr 1999.
- [10] J. Kleinberg, S.R.Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. *The web as a graph: Measurements, Models and methods*. In proceedings of the International Conference on Combinatorics and Computing, number 1627 in LNCS
- [11] Kleinberg 98. J. Kleinberg. *Authoritative sources in a hyperlinked environment*, Proc. 9<sup>th</sup> ACM-SIAM, 1998.
- [12] R. Lempel and S. Moran. *The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect*. The Technion, haifa 32000.
- [13] Andrew Marlett. *Web Rings Emerge as Alternative to Search Engines*. October 20, 1997.

- [14] Martindale and konopka 96. C. Martindale and A K Konopka. *Oligonucleotide frequencies in DNA follow a Yule distribution*, computer & Chemistry, 20(1): 35-38. 1996.
- [15] Neel Sundaresan. *Mining the web for relations*. In proceedings of the 9<sup>th</sup> International World Wide Web Conference, 2000.
- [16] Pirolli, Pitkow, and Rao 96. P. Pirolli, J. Pitkov, and R. Rao. *Silk from a sow's ear: Extracting usable structure from the web*, Proc. ACM SIGHI, 1996.
- [17].Pareto 1897.v.Pareto. *Cours d'economie politique, rouge, Lausanne et Paris, 1897*.
- [18] Simon 55. H. A. Simon. *On a class of stew distribution functions*, Bibliometrika, 42:425-440, 1995.
- [19] White and McCain 89. H.D. White and K. W. McCain, *bibliometrics*, in Ann. Rev. info.sci. and Technology, Elsevier, 1989, pp. 119-186.
- [20] Yule44. G. U. Yule. *Statistical study of Literary Vocabulary*, Cambridge University Press, 1944.
- [21] Zipf 49. G. K. Zipf. *Human Behavior and the Principle of Least Effort*, Addison-wesly, 1949.