Extract and Rank Web Communities

Asif Salekin
asalekin@gmail.com

Jeniya Tabassum Binte Jafar
jeniya.tabassum@gmail.com

Masud Hasan
masudhasan@cse.buet.ac.bd

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka-1000, Bangladesh

ABSTRACT

A *web community* is a pattern in the WWW which is understood as a set of related web pages. In this paper, we propose an efficient algorithm to find the web communities on a given specific topic. Instead of working on the whole web graph, we work on a web domain, which we extract based on the topic specific search results. Therefore, the resulted communities are highly related with the search topic.

The *ranking of a community* denotes the degree of relevance between the search query and the extracted communities. We introduce an approach for ranking the extracted communities based on their dense bipartite pattern. Ranking significantly improves the relevance of the extracted communities with the search topic.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering,Information filtering, Query formulation, Retrieval models, Search process, Selection process; H.3.5 [**Online Information Services**]: Data sharing, Web-based services

General Terms

Theory

Keywords

Web community, Ranking web communities, Structured web search, Dense bipartite graph, Web graph, Domain graph.

1. INTRODUCTION

The World Wide Web (WWW) contains a lot of web pages related to various types of information. Moreover, everyday new web pages are added on various new topics. Conventional search engines are used to find web pages related to the search query topics. However, sometimes it becomes difficult for the users to find the appropriate result of their search query from the large collection of resultant web pages provided by the search engine. To make this job easier, search engines apply several techniques to display them in more relevant and user friendly ways [4, 5, 9].

A common technique is to rank the web pages and to show the higher ranked pages at the top of the display [3, 6, 21]. Another well studied technique is to present the pages in a structured way by dividing them into some groups like images, video, webs and so on, where the groups are selected based on the page content [2, 14, 23].

Apart from grouping the web pages by their content, there is another way to group the web pages by communities [11–13, 15, 20, 22], where a single community contains the pages that are more "relevant" to each other. The meaning of "relevant" decides how a community would be defined. For example, in a web search, among the web pages that contain the search key(s), the pages whose links contain similar set of pages can define a community.

However, extracting communities from a vast number of pages is not an easy task. Previous approaches configured WWW as a web graph, where each page is a node and a link from one page to another defines a directed edge [7, 19]. In a web graph, the nodes having links with similar destinations are grouped in "clusters". Then each cluster represents a community [22].

Example of a community can be seen in Figure 1, which has been discovered from the search query of "laptop". Here the "hub" pages are of blogs and articles about various brand laptops, and "authoritative" pages are of various laptop sellers. To get information about the term "laptop", one can go to the "hub" pages to know about laptops, and then go to the "authoritative" pages to know the recommendation of the previous laptop users. This approach of finding web communities can also be applied in "web advertising". For advertising of laptop, the manufacturer company can find the communities of "laptop" as found in Figure 1 and then set their advertises in the "hub" pages of the higher ranked communities.

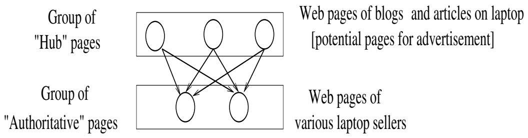


Figure 1: Community for the search query "Laptop"

1.1 Previous Works

Web structure mining discovers useful and informative patterns from the hyperlink structure of the web [18, 24]. Web community is one kind of variation of web structure mining. Several approaches have been developed to extract the web communities from the web graph [8, 10–13, 15, 20, 22]. The basic difference among these approaches is the notion by which a community is defined. There are two types of notions to define the web community: *cut based notion* and *bipartite graph based notion* [15].

WIMS '13, June 12-14, 2013 Madrid, Spain
Copyright ©2013 ACM 978-1-4503-1850-1/13/06 ...$10.00.

In the cut based notion, a community is defined as the set of web pages which have more links to the member pages of the community than to the non-members. Flake and colleagues [11, 12] proposed an algorithm which is based on maximum flow minimum cut theorem to extract such communities from the web graph.

The bipartite graph based notion comes from HITS (Hyper-link-Induced Topic Search) algorithm, which was proposed by Kleinberg [17]. Gibson et. al. [13] defined community as group of authoritative pages which have most number of hub pages pointed to them. Later, Kumar et al. [20] defined the *core* of a community as a group of pages that forms a bipartite clique [16] and proposed a trawling method to extract all the complete bipartite graphs which are considered as the cores of the communities. Such a core is used to find the rest of the community from the web. This approach is better than the approach of Gibson et. al. [13], as it requires that in the core of community all the authoritative pages will be pointed by all the hub pages. However, sometimes its requirements are too high to discover some specific communities [25]. Reddy and Kitsuregawa [22] proposed to relax the criteria of complete bipartite graph and proposed a *dense bipartite graph (DBG)* based approach. They first gather the related pages from the web graph and then extract the communities by extracting the DBGs from the gathered pages.

1.2 Our Approach

In this paper we introduce a novel approach to find web communities on a given specific topic. We use the results of specific query of a search engine (such as “bing”, “google”, etc.) to find web communities related to that topic. We crawl the resultant pages to find the possible domain of that search topic. Within this domain all the parent pages are potential hub pages and all the child pages are potential authoritative pages, and they form a bipartite graph. We propose an algorithm to find DBG pattern from this extracted web domain and thus we find the query specific web communities.

We also propose an approach to rank the extracted communities, where we use two factors–the *density* of the community and the *ratio* of hub and authoritative pages in the community. The higher ranked communities have more relevance with the query topic than the lower ranked communities.

The rest of the paper is organized as follows. In Section 2, we discuss some preliminaries. In Section 3, we discuss the DBG based approach and the scope to improve on it. In Section 4, we discuss our proposed approach to extract web communities from the web domain based on a specific search query topic. In the same section, we also discuss the ranking of the extracted web communities. In Section 5 we show our experimental results. Finally, Section 6 concludes the paper with some scopes for future works.

2. PRELIMINARIES

For a topic specific community search, we use a notion called *domain graph*, which is defined as follows. We extract some web pages form the search query results of a search engine. We crawl the children pages and parent pages for each of these extracted pages. Then our domain graph consists of these extracted and crawled pages and hyper-links between them.

Size of a domain graph is big. If the number of extracted pages is n and if the number of children and parent pages are n_c and n_p, respectively, then the total number of nodes in our domain graph is $n(1 + n_p + n_c)$.

We define a *community* by two groups of web pages, called *hubs* and *authoritative pages*, that are related to a specific search query. These hubs and authoritative web pages are related by a DBG pattern. A community is defined as a group of nodes that form a $DBG(T, I, p, q)$, where T denotes the set of hubs, I denotes the set of authoritative pages, p denotes minimum number of outgoing links from n_i, $n_i \in T$, and q denotes minimum number of incoming links to n_j, $n_j \in I$. The corresponding CBG(Complete Bipartite Graph) pattern of a DBG, can be expressed by $DBG(T, I, n_I, n_T)$ where n_I denotes total number of nodes in I and n_T denotes total number of nodes in T. Such DBG patterns are extracted from a set of web pages, which we have found using the resulted pages of search query and the web pages pointed by the incoming and outgoing hyper-links of those resulted web pages. Previously [22] defined community considered only the hub pages. But in our approach we consider both the hub pages and authoritative pages since we get query related information from the nodes of both sets T and I. This proposed notion of web community is described in details in Section 4.

An *edge weight* is defined as the number of hyper-links from a parent node (hub page) to a child node (authoritative page). Suppose there are i number of hyper-link from n node (web page) to m node (web page). Then the weight of the edge from node n to node m will be i. Higher weight of the edge from a node n to node m means that node n is strongly referring node m.

The following fact is important in our results.

Fact 1: In a particular domain graph, for a specific value of p and q every web page belongs to one distinct community.

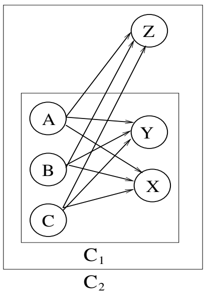


Figure 2: Illustrating Fact 1.

An illustration of Fact 1 is as follows. Also see Figure 2. Suppose, C_1 is a distinct community in a domain graph G, where the values of p and q are specified. In C_1, A, B, C are three random nodes in T and X, Y are two random nodes in I. For each node in T, there are p outgoing edges to the nodes of I and for each node in I there are q incoming edges from T. Let, C_2 be another distinct community in the same domain graph G, where values of p and q are identical to that in community C_1. The set T of C_2 is also identical to C_1, but the I set is different. Suppose, X, Y, Z are three random nodes in I set of C_2. So, Z has at least q incoming edges form the nodes of T. According to the definition of community, Z would be a member of I set of C_1 also. Hence, C_1 and C_2 can not be two distinct community.

3. DBG BASED APPROACH AND SCOPE OF IMPROVEMENT

In [22], a community is considered as a set of closely related pages that form a DBG. From a given data set of web pages, the algorithm in [22] finds a DBG structure of web pages which is considered as a community based on the relax cocite relationship. The algorithm they proposed is explained below.

3.1 The DBG Algorithm

The algorithm in [22] has two phases: *gathering related pages* and *DBG extraction*. In the first phase, the algorithm starts with one node, which is considered as the initial parent node of set T. Then in every iteration, for each node in T, all the child nodes are gathered and added to the child set I. After that, for each node in I, all the parent nodes are gathered and added to the parent set T. This process is continued as long as it is possible to gather parents and children. The gathered parent and child nodes and the corresponding edges from parents to children constitute the *gathered graph*.

The DBG extraction phase starts with selecting the value of both p and q. In every iteration, all the nodes in T having outbound links less then p are discarded. Similarly, all the nodes in I having inbound links less then q are discarded. This process is continued until the gathered graph converges to a DBG pattern of community. The remaining nodes represent a community. By increasing the values of p and q the algorithm [22] finds the higher level communities.

3.2 Illustration and Scope of Improvement

According to [22], a *single* community is represented by a *single* DBG pattern. However, it can be found that after the pruning is executed in the DBG extraction phase, a single communtiy may contain more than one DBG pattern, which is an inconsistency with the definition.

For an example, suppose that after the first phase of the algorithm in [22] we have $T = \{A, B, C, P, Q, T\}$ and suppose that the graph is like that in Figure 3(a). So, in the pruning phase, the edges will be $\{< A, F >, < A, D >, < B, F >, < B, D >, < C, F >, < C, D >, < C, E >, < P, E >, < P, R >, < P, S >, < Q, R >, < Q, S >, < T, R >, < T, S >\}$. When the DBG extraction method is applied with $p = 2$ and $q = 3$, we will get the community $C = \{A, B, C, P, Q, T\}$ and the community graph will be like that in Figure 3(b). However, there are in fact two DBG and the corresponding two communities should be: $C_1 = \{A, B, C\}$ and $C_2 = \{P, Q, T\}$. Therefore, the member nodes of C should belong to two distinct communities instead of one.

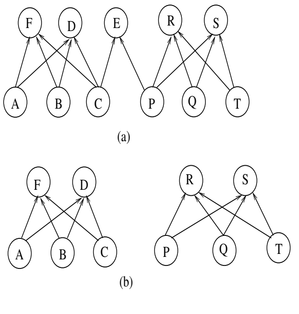


Figure 3: (a) Nodes gathered in the phase of gathering related pages. (b) Communities founded after the DBG extraction phase.

4. OUR PROPOSED APPROACH

We present an approach to extract communities related to a specific topic. Our community extraction algorithm extracts communities from the domain graph, which we create based on the results of search engine query. We also propose an approach to rank the extracted communities based on their density and hyperlink properties. Our approach has three steps: (1) domain graph construction, (2) extraction of communities from domain graph, and (3) rank extracted communities of specific p, q values.

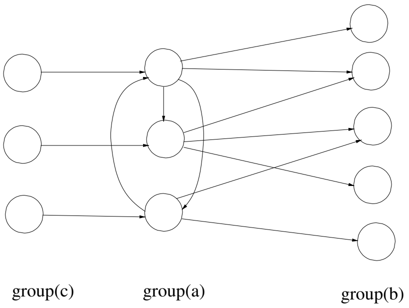


Figure 4: Pages of group (a) are the initial N pages of the search query results. Pages of group (b) are found by crawling group (a) pages (which are not in group (a)). Pages of group (c) are the parent pages of group (a) (which are not in group (a)).

4.1 Domain Graph Construction

In our approach we construct the domain graph related to a specific search query topic using the search results of a search engine. For our experiment we use “bing” as the background search engine.

Suppose we are willing to find communities of search query “A”. We take first N search results from the search engine for the query “A” and call these pages *initial pages*. Web pages found as the search query result can be both potential hub pages and authoritative pages of a community. A potential hub page has outbound links to potential authoritative pages. All the potential authoritative pages may not be found in the initial pages. So, we crawl each of these initial pages to find all the potential authoritative pages. Again, a potential authoritative page is pointed by many potential hub pages. Now all the hub pages pointed to the resulted authoritative pages may not be found in the initial pages. So we find the parent web pages of each initial page to gather all the potential hub pages. We use “bing” for this purpose.

Figure 4 illustrates how we have used crawler and “bing” to retrieve domain graph. From this domain graph we extract community structures. First we find the N initial pages of group (a). These N initial pages are the first N results of the specific search query. Then we crawl the web pages of group (a) and find two types of pages: (1) some of the outbound links may refer to pages of group (a) and (2) other outbound links will refer to the web pages not in group (a).

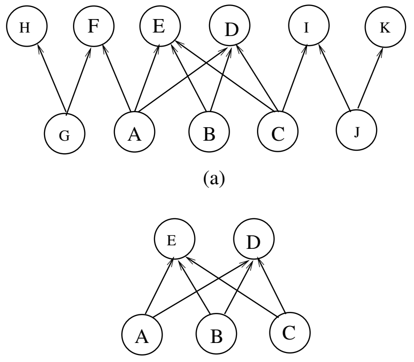
The second group of web pages are included in group (b). These web pages are potential authoritative pages (not present in the initial N resulted web pages). After that we use “bing” to find parent pages of all the N initial resultant pages. There are two types of pages among these parent pages: (1) some of the parent pages are present in the initial N resulted pages (pages of group (a)), and (2) other parent pages are not included in the search results (pages of group (c)). The Pages of group (c) are potential hub pages.

There can be links between pages of group (a) to group (a), pages of group (a) to group (b) and pages of group (c) to group (a). As the pages of group (a) have both inbound and outbound links, they can be both hub and authoritative pages. But web pages of group (b) have only inbound links. So, they can only be authoritative pages. Finally, web pages of group (c) has only outbound links, so they can only be hub pages.

4.2 Community Extraction

Given the collection of nodes (web pages), our approach is to extract communities for specific p and q values. The input of the algorithm is the N initial pages we found as the search results of the specific search query topic and the value of p and q. The output contains dense bipartite graphs, such as $DBG(T, I, p, q)$. The pseudo code of this approach is stated in Algorithm 1.

Our algorithm does gathering and pruning simultaneously. While gathering the potential nodes of a community, when we find a father node having less than p child nodes, or a child node having less than q father nodes than we prune that node. Consider Figure 5(a). Suppose the algorithm starts with node A. Later while evaluating node F the algorithm finds that F has less than q father nodes ($p = 2$, $q = 3$), hence F is discarded from the set "CHILD PAGES". So, the algorithm does not need to consider nodes G, H. Similarly the algorithm discards node I from the set "CHILD PAGES" and does not need to consider nodes J, K. But the previously proposed algorithm of [22] considers all the nodes of Figure 5(a) in the"Gathering related pages" phase. Hence our proposed algorithm consumes less memory and time, though both algorithms extract same community of Figure 5(b).



(a)

(b)

Figure 5: (a) Domain graph (b) Communities extracted from Domain graph

Our proposed algorithm tries to extract community for every initial N pages. Hence, for a specific topic with N initial pages there may be several resulted communities. But for a single page of N initial pages the proposed algorithm will extract a single DBG pattern (if exists), which is considered as a community. Consider Figure 3. Suppose we start our algorithm with the father node A. Later the algorithm gradually evaluates nodes F, D, B, C, E. While evaluating node E, since it has less than q father nodes (here p=2, q=3), it is discarded from the set "CHILD PAGES". Thus "CHILD PAGES" and "FATHER PAGES" sets are converged and we found the community C_1. Similarly, when the algorithm starts with node P, Q or T, it finds the community C_2. So, starting with a single node our algorithm extracts a single community which always consists of a single DBG.

According to fact-1, in a domain graph, a node can be a member of a single community. So, when we found a father node as the member of any previously found community we do not need to proceed the algorithm for that father node any more. So, inclusion of fact-1 reduces run time significantly.

```
input  : Set of nodes(web pages) $W$, $p$, $q$
output: Set of communities in the form of $DBG(T, I, p, q)$
for every page $S_i$ of input set $W$ do
    if number of children pages of $S_i \geq p$ then
        insert child pages of $S_i$ in the set "CHILD PAGES";
    end
    else
        NO community found for $S_i$ ;
    end
    mark $S_i$ as seen;
    repeat
        for every page $C_i$ of the set "CHILD PAGES" do
            if $C_i$ is not marked as seen then
                if number of father pages of $C_i \geq q$ , in the
                "Domain Graph" $G$ then
                    insert father pages of $C_i$ in the set
                    "FATHER PAGES" ;
                    mark $C_i$ as seen;
                end
                else
                    delete $C_i$ from the set "CHILD PAGES",
                end
            end
            else if $C_i$ is marked as seen then
                if number of father pages of $C_i$ is less than $q$
                ,in the set of "FATHER PAGES" then
                    delete $C_i$ from the set "CHILD PAGES";
                end
            end
            else if $C_i$ is in any previously found communities
            for the same value of p, q then
                No new community found for $S_i$
            end
        end
        for every page $f_i$ of the set "FATHER PAGES" do
            if $f_i$ not marked as seen then
                if number of children pages of $f_i \geq p$ , in the
                "Domain Graph" $G$ then
                    insert child pages of $f_i$ in the set "CHILD
                    PAGES" ;
                    mark $f_i$ as seen;
                end
                else
                    delete $f_i$ from the set "FATHER PAGES";
                end
            end
            else if $f_i$ is marked as seen then
                if number of child pages of $f_i < p$ in the set
                "CHILD PAGES" then
                    delete $f_i$ from the set "FATHER PAGES";
                end
            end
            else if $f_i$ is in any previously found communities
            for the same value of p, q then
                No new community found for $S_i$
            end
        end
    until the sets "CHILD PAGES" and "FATHER PAGES"
    are converged;
end
```

Algorithm 1: Extracting Web Communities

4.3 Ranking

According to our proposed notion of community, we can say that if the DBG structure of the community is close to its corresponding CBG structure, then the pages of the community are highly interlinked. Therefore that would be a better community. To rank the community we consider the edge weight among two nodes. Higher edge weight denotes that the child node is highly referenced by the parent node. So we use ratio r_1 in ranking as follows,

$$r_1 = \frac{\sum edge\ weight\ in\ the\ extracted\ DBG}{number\ of\ edges\ in\ the\ corresponding\ CBG} \tag{1}$$

A higher value of r_1 refers to higher interlinks and strong references between hub and authoritative pages. Again, from our proposed notion of the community, if an authoritative node is referenced by many hub nodes then the community is better. For an example, consider two communities C_1 and C_2, where in community C_1, the number of hub nodes are 5 and the number of authoritative nodes are 3, and in community C_2, the number of hub nodes are 4 and the number of authoritative nodes are 4. Then we will consider C_1 to be a better community than C_2. Formally, we introduce the r_2 ratio in ranking as follows,

$$r_2 = \frac{number\ of\ nodes\ in\ T}{number\ of\ nodes\ in\ I} \tag{2}$$

Here, T is the set of Hub nodes and I is the set of Authoritative nodes. A higher value of r_2 refers to higher hub-authoritative ratio. The final ranking (r) of the community is represented as

$$r = r_1 * r_2 \tag{3}$$

Higher value of r refers to the better communities. Here, we use the multiplicative measure. Because from our experiment we find that, multiplicative measure gives better result than additive measure.

5. Experimental Results

In this section we present the experimental results that we conducted on our proposed approach. We have implemented our algorithm in php and run a preliminary simulation on a 2.3 GHz PC with Intel core i5 processor and 4 GB memory. We have uploaded our code, which is free to access, at [1].

5.1 Domain Graph Collection

In our experiment while collecting the initial pages we use $N =$ 100, 150 and 200. Then the constructed domain graph is stored in MySQL database. In this phase we exclude some of the web pages from our consideration (pages come from the outbound links of the search result), as they can never be a potential authoritative page. For example the pages like: "terms and conditions", "policies", "privacy", "copyright", "contact us", "general disclaimer", "advertising overview", etc.

5.2 Community Extraction Phase

For each topic specific community search, our approach extracts communities from domain graph for specific p and q values. While conducting the experiment we initially set $p = 2$ and $q = 3$, and extract the communities. Then we increase the value of the p and q by one and again conduct the whole community extraction process to find the denser communities. We continue this increment process until no more community is found.

5.3 Findings

We run our experiments for 30 search queries. From that experimental results here we show 6 random examples.

In Table 1 and Table 2 we show some random results of the proposed community search algorithm. These tables include the size of domain graph, number of extracted communities, size and rank of the best community for specific p and q values, and the percentage of relevance of community's nodes with that query topic. Results in these tables reflect that the higher value of p, q results in a higher ranked community. This is because when the value of p, q increases the density of interlinks between the nodes of T and I increases. As a result value of the parameter r_1 [equation 1 in section 4.3] increase which increases the rank of extracted communities. Also, with the increase of p, q value the number of communities reduces. Because when the value of p, q increases the loosely related communities are pruned. Table 1 and 2 also show that higher ranked communities have more relevance with the query topics. Because the higher the rank of community the denser it is. And denser communities have less irrelevant nodes. That is because most of the topic irrelevant nodes are loosely connected with the nodes of set T and I. And such nodes are discarded with the increase of p, q value. So, higher value of p, q refers to higher ranked and highly relevant communities.

Consider the example of "Xbox 360" in Table 1. When we take first 150 pages from the query result we get a domain graph with 10927 web pages. Initially we set $p = 2$ and $q = 3$ and extract 2 communities from the domain graph where the best community has rank value, $r = 0.049$ and 82.5% nodes of the communities contain information related to "Xbox 360". Later we gradually increase the value of p, q and get higher ranked communities with more relevant nodes. While increasing their values when we set $p = 5$, $q = 6$ the number of communities are reduced to 1, since only highly dense community exist for higher value of p, q. This is because then the irrelevant nodes are discarded from the sets T and I. So the loosely related communities are pruned. We increase the value of p and q until 10 and 11 since there exists no community for $p = 11$, $q = 12$. With increase of value of p, q the rank of best ranked community increase from 0.049 to 0.248 and relevance of nodes in communities increase from 82.5% to 91.3%. Hence for search query topic "Xbox 360", we find higher ranked and more related communities for higher value of p, q.

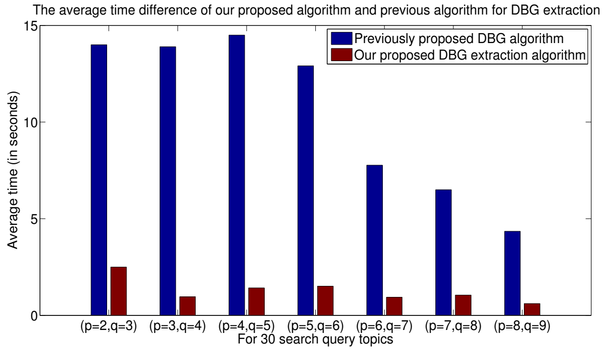


Figure 6: Comparing run time with previous algorithm

As we simultaneously execute gathering and pruning the run time reduces significantly. To show this improvement of run time, we apply our algorithm and the previously proposed algorithm of

Query	Number of search query results	Number of pages in the Domain Graph	p	q	Number of Communities	Best Community Rank	Best Ranked Community Size		Relevance of nodes in communities
							T	I	
Xbox 360	100	8799	2	3	2	0.042	11	53	82.8%
			3	4	1	0.201	14	78	83.6%
			4	5	1	0.213	14	65	84.8%
			5	6	1	0.218	10	51	85.2%
			6	7	1	0.231	9	46	87.2%
			7	8	1	0.238	5	37	90.4%
	150	10927	2	3	2	0.048	15	76	82.5%
			3	4	2	0.049	5	11	81.2%
			4	5	1	0.197	18	89	85.1%
			5	6	1	0.200	18	88	85.9%
			6	7	1	0.216	18	84	88.2%
			7	8	1	0.216	18	84	88.2%
			8	9	1	0.241	14	76	90%
			9	10	1	0.242	11	56	91%
			10	11	1	0.248	7	51	91.3%
	200	12109	2	3	2	0.043	6	19	80%
			3	4	2	0.051	5	13	83.3%
			4	5	1	0.239	21	102	84.5%
			5	6	1	0.281	21	99	85.8%
			6	7	1	0.298	19	92	86.4%
			7	8	1	0.305	19	88	88.7%
			8	9	1	0.312	18	85	89.3%
			9	10	1	0.340	11	79	90%
			10	11	1	0.358	9	63	90.2%
Alan Turing	100	5257	2	3	1	0.09	92	68	92.5%
			3	4	1	0.40	19	17	93.4%
	150	8910	2	3	1	0.07	136	106	91.9%
			3	4	1	0.21	35	38	92.5%
			4	5	1	0.40	16	21	94.8%
			5	6	1	0.78	9	11	95.6%
	200	10793	2	3	1	0.05	158	145	90%
			3	4	1	0.17	41	47	91.4%
			4	5	1	0.41	19	22	92.6%
			5	6	1	0.60	13	15	94.8%
Hero Honda	100	4205	2	3	4	1.56	4	3	82.5%
			3	4	1	0.72	4	6	83.6%
	150	7383	2	3	7	1.56	4	3	82.5%
			3	4	2	4	5	4	83.1%
	200	10472	2	3	9	3.5	3	2	80.4%
			3	4	4	2.04	6	7	83.3%
			4	5	1	0.82	5	7	84.5%
Job Bangladesh	100	6744	2	3	1	0.128	43	21	93.6%
			3	4	1	0.221	35	13	93.7%
			4	5	1	0.232	21	9	96.6%
	150	7320	2	3	2	2.5	5	2	94.7%
			3	4	1	0.404	65	28	96.2%
			4	5	1	0.526	46	21	97.01%
			5	6	1	1.083	7	6	100%
	200	9788	2	3	2	2.5	5	2	93.2%
			3	4	1	0.44	89	43	94.6%
			4	5	1	0.57	74	31	95.2%
			5	6	1	0.93	33	7	97.5%

Table 1: Performance of community search for the queries- Xbox 360, Alan Turing, Hero Honda, Job Bangladesh

Query	Number of search query results	Number of pages in the Domain Graph	p	q	Number of Communities	Best Community Rank	Best Ranked Community Size		Relevance of nodes in communities
							T	I	
Nokia Asia	100	7349	2	3	1	0.182	29	45	83.7%
			3	4	1	0.475	15	11	84.6%
			4	5	1	1.085	8	7	86.6%
	150	8128	2	3	1	0.128	40	50	84.4%
			3	4	1	0.438	19	16	94.28%
			4	5	1	1.061	9	7	100%
	200	10329	2	3	1	0.141	60	67	85.03%
			3	4	1	0.395	32	49	92%
			4	5	1	0872	15	9	92%
Ubuntu	100	7980	2	3	1	0.019	78	187	93.9%
			3	4	1	0.049	11	132	94.4%
			4	5	1	0.054	5	113	95.7%
	150	9215	2	3	1	0.021	118	254	93%
			3	4	1	0.041	11	147	94.9%
			4	5	1	0.043	6	139	95.8%
			5	6	1	0.045	6	132	96.3%
	200	11239	2	3	1	0.022	139	312	88.9%
			3	4	1	0.048	108	243	91.4%
			4	5	1	0.087	22	183	92.6%
			5	6	1	0.157	13	143	94.8%

Table 2: Performance of community search for the queries- Nokia Asia, Ubuntu

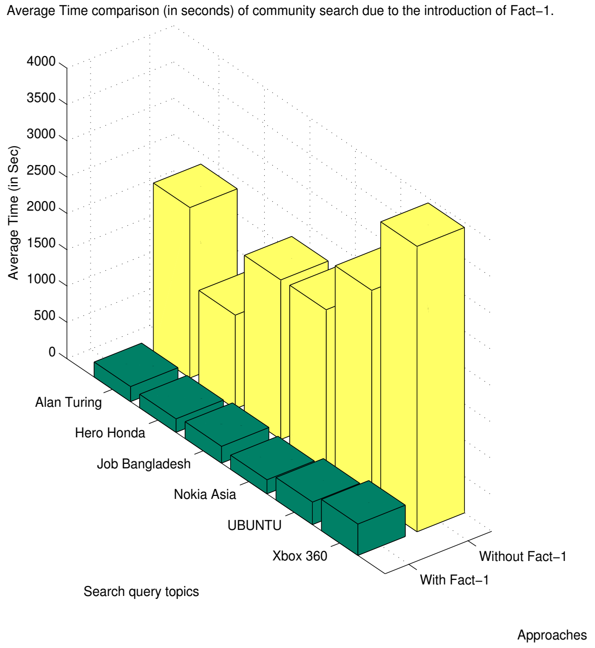


Figure 7: Improvement in run time for introducing Fact 1

[22] on the same domain graphs. Figure 6 shows the average time difference of the two algorithms to find a community starting with a single initial node. This illustration reflects that, the average time reduction for 30 queries is- 80.7% when $p = 2$, $q = 3$; 90.4% when $p = 3$, $q = 4$; 85.7% when $p = 4$, $q = 5$; 84% when $p = 5$, $q = 6$; 86.7% when $p = 6$, $q = 7$; 83.3% when $p = 7$, $q = 8$;

90% when $p = 8$, $q = 9$.

It should be noted that we introduced Fact 1 in our approach. Without this fact we have to execute community extraction procedure for all of the web pages of group (a) of Figure 4. But in this procedure similar communities are extracted more than once. Using Fact 1 we can avoid this repetition that would reduce the run time of community extraction significantly. For the 30 queries in our experiment, the average run time reduction is about 90%. Figure 7 reflects the reduction of time in 6 random search queries due to introduction of the Fact 1.

6. CONCLUSION

In this paper, we propose an approach to extract communities related to a specific search query topic and represent the communities in a rank based notion. We take first few resulted pages on the query topic from a search engine and then make a "Domain Graph" crawling these resulted pages. We propose an algorithm to extract the clusters from this domain graph. These clusters represents the communities on the search query topic. Then we use a ranking approach to rank the extracted communities to show the search query result in a structured way.

Our approach was based on bipartite notion of community. This approach can be extended to find communities defined by other notions, which may give more relevant and structured search results.

There are many other patterns in WWW. If we can extract and integrate these patterns with web search then various types of structured search can be introduced.

7. REFERENCES

[1] `https://gist.github.com/44acb9783696bc8e33e4/`.

[2] G. Attardi, A. Gulli, and F. Sebastiani. Automatic web page categorization by link and context analysis, 1999.

[3] R. Baeza-Yates. Web page ranking using link attributes. In *Proceeding of the 13th International Conference on World Wide Web (WWW'04)*, pages 328–329, New York, NY, USA, May 2004.

[4] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21th SIGIR*, pages 104–111, 1998.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107–117. Elsevier, 1998.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107–117. Elsevier, 1998.

[7] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33(1-6):309–320, 2000.

[8] G. Buehrer and K. Chellapilla. A scalable pattern mining approach to web graph compression with communities. In *Proceedings of the international conference on Web search and web data mining (WSDM '08)*, pages 95–106. ACM, 2008.

[9] J. Dean and M. R. Henzinger. Finding related pages in the world wide web. *Computer Networks*, 31(11–16):1467–1479, 1999.

[10] Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In *Proceedings of the 16th international conference on World Wide Web(WWW '07)*, pages 461–470. ACM, 2007.

[11] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, Boston, MA, USA, August 2000.

[12] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–71, 2002.

[13] D. Gibson, J. M. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space - Structure in Hypermedia Systems*, pages 225–234, Pittsburgh, PA, USA, June 1998.

[14] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pennock, and G. W. Flake. Using web structure for classifying and describing web pages. In *Proceedings of the 11th International Conference on World Wide Web (WWW'02)*, pages 562–569, Honolulu, Hawaii, USA, May 2002.

[15] J. Han, X. Hu, and N. Cercone. On graph-based methods for inferring web communities. In *Proceedings of the Workshop on Applications, Products and Services of Web-based Support Systems (WSS'03)*, pages 145–152, Halifax, Canada, 2003.

[16] D. S. Hochbaum. Approximating clique and biclique problems. *J. Algorithms*, 29:174–200, 1998.

[17] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[18] R. Kosala. Web mining research: A survey. *SIGKDD Explorations*, 2:1–15, 2007.

[19] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. The web as a graph. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 1–10, Dallas, Texas, USA, May 2000.

[20] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. *Computer Networks*, 31(11–16):1481–1493, 1999.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[22] P. K. Reddy and M. Kitsuregawa. An approach to relate the web communities through bipartite graphs. In *Proceedings of the 2nd International Conference on Web Information Systems Engineering (WISE'01)*, pages 301–310, Kyoto, Japan, December 2001.

[23] S. Sclaroff. World wide web image search engines. Technical report, Proceedings of NSF Workshop on Visual Information Management, 1995.

[24] J. Srivastava and R. Cooley. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1:12–23, 2000.

[25] K. Verbeurgt. Inferring emergent web communities. In *Proceedings of the International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine, and Mobile Technologies on the Internet*, 2003.