ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

Implementing Dynamic Authority Based Search Using Binrank Algorithm

Silica Kole[1], Ashish Kumar[2], Pranav Bhatia[2], and AnshulGoel[2]

Associate Professor, Department of CSE, Bharti Vidyapeeth's College of Engineering, GGSIPU University, New Delhi, India[1]

Department of CSE, Bharti Vidyapeeth's College of Engineering, GGSIPU University, New Delhi, India[2]

ABSTRACT: Various algorithms such as ObjectRank and PageRank, the latter created by Larry Page and used in Google Search Engine, were highly expensive as they required a PageRank-style iterative computation over the full graph. BinRank, a hybrid algorithm proposed uses an index of pre-computed results for some/or all keywords being used by the user. It approximates ObjectRank result by materializing a relatively smaller subsets of data graphs, which are generated by partitioning all the terms in the corpus based on their co-occurrence. BinRank can achieve sub-second query execution time without affecting the quality of the search results.

KEYWORDS: Approximation algorithms, Material Sub-graphs (MSG), Online Keyword search, Bins

I. INTRODUCTION

Before BinRank, algorithms such as PageRank and ObjectRank were used for searching of information over the internet. The PageRank algorithm [1] utilized the Web graph link structure which was then used to assign a global importance to Web Pages. PageRank followed the outgoing links with uniform probability and the score was independent of the query. A more dynamic approach of PageRank, known as Personalized Page Rank (PPR) was developed due to the increasing demand for greater flexibility in ranking of the web pages. The basic idea of PPR is simple: instead of distributing the sources of PageRank uniformly, the distribution is improved by modifying it according to each individual query [2]. For a given preference set, it performs an expensive fix-point and iterative computation over the complete Web graph. Thus the issue of scalability is a huge disadvantage of this algorithm. ObjectRank extends the PPR to perform the keyword search in databases rather than the Web graph. Another novelty of ObjectRank is that it is not global as google, i.e., for each <keyword, object> we compute an ObjectRank value [3]. It has been successfully applied on databases having social networking components such as collaborative product design andbibliographic data. But unfortunately, ObjectRank suffers from the same scalability issues as that of PPR, since it requires various iterations over all links and nodes of the entire database graph [4]. The ObjectRank has 2 modes: Offline and Online. The Online mode runs the ranking algorithm only when the query is received, which takes up too much time on long graphs. For example, on a graph of articles of English Wikipedia with 3.2 million nodes and 109 million links, even a fully optimized in-memory implementation of ObjectRank takes 20-50 seconds to run [4]. In the Offline mode, the top-k results are computed in advance for a query. This is quite expensive in terms of storage space and is not suitable for the terms outside thequery workload that the user may search for. Therefore, in this paper we have introduced a hybrid approach called *BinRank*where the query accessing time can be traded off with storage and preprocessing time. BinRank closely approximates the ObjectRank scores as it applies ObjectRank on a smaller subgraph rather than the complete graph. These subgraphs are computed offline in advance and are stored in memory. The precomputation can be parallelized with linear scalability. An example of full Wikipedia data set is taken by us where BinRank can answer any query in less than 1 second, by precomputing about a thousand subgraphs, which takes only about 12 hours on a single CPU. While deciding on the precomputation of the subgraph, there are usually two problems faced: a) the number of subgraphs to compute b) the construction of each subgraph. The intuition behind our approach is that a subgraph that contains all objects and links relevant to a set of related terms should have all the information needed to rank objects w.r.t. one of these terms. For 1), we group all terms into a small number (around 1,000 in case of Wikipedia) of "bins" of terms based on their co-occurrence in the entire data set. For 2), we execute ObjectRank for each bin using the terms in the bins as random walk starting.

II. OBJECTIVE

The objective of the paper is to:

- Approximate ObjectRank by using Materialized SubGraphs (MSGs), which can be precomputed offline.
- Use of ObjectRank itself to generate MSGs for “bins” of terms.
- Introduce a greedy algorithm that minimizes the number of bins by clustering terms with similar posting lists.

III. OBJECTRANK-BACKGROUND

A. *Data Model*

Unlike PageRank, ObjectRank performs top-k relevance search over a database rather than a Web Graph. The data graph G (V, E) is used to represent the objects and the semantic relationships as nodes and edges, where edges represent the hyperlinks between Webpages in a PageRank. A node v ϵ V contains a set of keywords and its object type. For example, when a paper u cites another paper v, ObjectRank includes in E an edge e = (u→v) that has a label “cites.” It can also create a “cited by”—type edge from v to u . By assigning different edge weights to different edge types, ObjectRank can capture important domain knowledge.

B. *Query Processing*

The query processing in ObjectRank uses Random Surfer Model [5]. The model starts from a random node vi among nodes that contain the keyword. The starting points are called a base set. For a keyword k, the keyword base set of k, BS (k), consists of nodes in which k occurs. Any node in Graph G can be a part of BS (k), which makes it support full degree of personalization. At each node, the surfer follows outgoing edges with a probability p, or jumps back to a random node in BS (k) with probability (1-p)2. At a node v, when an edge is determined to be followed, each edge e that is originated from v is chosen with probability w ((e)) /(OutDegree(λ(e),v)), where OutDegree (k ,v) denotes the number of outgoing edges of v whose edge types are similar to k. The score of vi is the probability of r (vi) that a random surfer is found at vi at a certain moment.

C. *Quality compared to PageRank*

The ObjectRank is in contrast with the PageRank approach which returns objects containing the keyword that is sorted according to their score. ObjectRank on the other hand, it utilizes the link structure that captures the semantic relationships between objects which is useful in showing even those object that don’t have the keyword but are highly relevant and thus can be included in the top-k list. This makes the ObjectRank of having a superior result quality.[6]

IV. BIN CONSTRUCTION

A set of MSGs for terms of a dictionary are constructed by partitioning them into a set of term bins, which is based on co-occurrence. We generate an MSG for every bin on intuition that the sub graph that contains all the objects and links which are relevant to the set of related terms must have all the information needed to rank objects with respect to one of those terms. There are two important goals while constructing a bin. 1) Controlling the size of each bin. 2) Minimizing the number of bins to save the preprocessing time. To achieve the first goal a maxBinSize parameter is introduced which limits the size of the union of the posting lists of the terms in the bin, called bin size. [7]

V. VARIOUS MODELS OF BINRANK SYSTEM

A. *User Registration*

To access the BinRank system, each user must be registered and must have an account. New user can create a new id and choose a password, along with other basic details such as Name, country, etc. For the User Registration we created a Registration.jsp. Every member must have a unique id and a password for moving further. The information of each user is stored in login table.

B. *Search Query Optimization*

Now after the user logins, he is given an option to select the Key words of the search he intends to make. For example, if a user wants to search for a beach in Chennai, or Temples in Mumbai, he selects the first Keyword, i.e., the Category from a drop down list and the next keyword by writing in the TextField and finally pressing the Search button.

C. *Index Creation*

When a user clicks on a link, an index is created in the PageRank table. The schema of the index is:

Link	Rank	Key1	Key2
www.mumbai.org/beaches	1	Temples	Mumbai

Figure 1: Schema of Index

Users can search any kind of things in our application when we connect with Internet. Users query will be processed based on their submission, and then it will produce the appropriate result. Result will be produced based on BinRank.

D. *BinRank Algorithm Implementation*

A MSG will be generated for each bin based on a intuition that a subgraph that has all objects and links relevant to a set of related terms. The information is needed to rank the objects with respect to each other. Based on this index generation, the results will also be generated of the users' query. BinRank algorithm will use the indexing and ranking techniques to produce the efficient results in short time.

E. *Graph Based on Rank*

A graph will be generated based on the users' queries that are submitted. This graph will represent the search key-word, number of websites produced for the search, the number of times that website occurred in the search result and the Rank of the website based on the user clicks. User may search the same key-word repeatedly, and the result may produce the same URLs. At that user will click some of the URLs; based on their clicks the Rank will be calculated. Based on the Number of times URL occurrence, Rank and Keyword the Graph will generate.

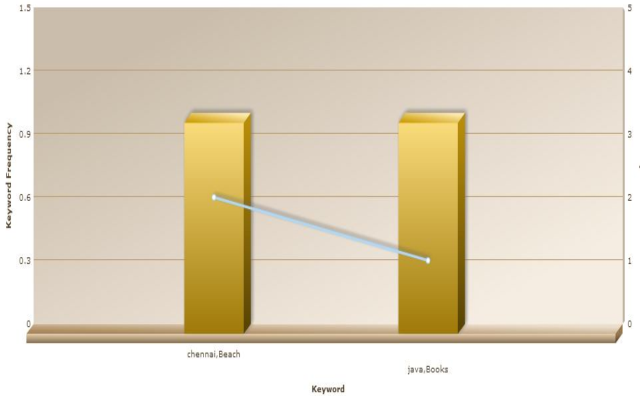

Figure 2: Graph created using BinRank

VI. CONCLUSION

In this paper, we have proposed BinRank as a better and practical solution for dynamic authority-based searching. The technique is based on partitioning and approximating various sub graphs. BinRank offers a nice tradeoff between query time, preprocessing time and storage. PageRank algorithm suffered from an expensive fix-point and iterative computation and scalability issues. ObjectRank extended the Personalized Page rank to perform search on databases rather than the Web graphs. Also it may/may not show only those objects that don't have the keyword but are highly relevant. It made it more superior but is suffered from the same scalability issues as that of PPR. BinRank applies ObjectRank on a smaller sub graphs rather than the compete graph, and these sub graphs are computed offline. A set of MSGs are constructed by partitioning them into set of term bins, which is based on their co-occurrence.

REFERENCES

1. S.Brin, L.Page,"The Anatomy of a Large-Scale Hypertextual
2. Web Search Engine",Computer Networks,vol.30, nos.1-7, pp. 107-117,1998
3. Sree Lakshmi Pinapatruni, Satya P Kumar Somayajulaa, “Generating MSG’s by Binrank for scaling in Dynamic Authority Based Searcg”, IJSCT Vol. 2, Isue 3, September 2011.
4. Andrey Balmin, VagelisHristidis, YannisPapakonstantinou, "ObjectRank: Authority-Based Keyword Search in Databases"Amjan. Shaik, Nazeer. Shaik, "Scale the Active Influence Based Investigation Using Materialized Sub Graphs",IJSCT Vol. 2 (3) , 2011, 1358-1363
5. V. Hristidis and Y. Papakonstantinou. DISCOVER: Keyword SearchIn Relational Databases. VLDB, 2002,
6. http://www.computer.org/csdl/proceedings/icde/2009/3545/00/3545a066-abs.html
7. http://www.scribd.com/doc/51322117/BIN-RANK