**Article Citation Format**

Yaya, O.J & Robert, A.B.C (2018). An Enhanced algorithm For Ranking Results Based on User Specified Traits. Advances in Mathematical & Computational Sciences . Vol. 6 No. 2. Pp 63-86.

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 13rd May, 2018
Review Type: Blind
Final Acceptance:: 19th September, 2018
DOI Prefix: 10.22624

# An Enhanced Algorithm For Ranking Results Based on User Specified Traits

**Yaya, J.O & Robert, A.B.C.**
Department of Computer Science
University of Ibadan
Ibadan, Nigeria

## ABSTRACT

Our intention is to develop a recursive ranking algorithm with wide area of application. Fundamental to the achievement of this goal is the formulation of a model that will adequately characterise objects in the problem domain. Such model enabled users find what they want based on individual bias or preferences. The model was translated into a recursive ranking algorithm that traversed the entire problem domain, and generated the most appropriate set of results.We developed an algorithm that allow users to specify search criteria in order of relative importance to their search for people, publications, records, etc. on the Web, social networks, citation databases, and so on. This was done by the systematic analysis of different models, algorithms and related works as covered in-depth in previous works. We then formulate our model – the Enhanced Recursive Ranking Algorithm (ERRA) and carried out normalization of the dataset that were used in our test results. Testing and presentation of results are presented

**Keywords**: Algorithms, Ranking, ERA, User Specified traits and Normalization.

## 1. BACKGROUND TO THE STUDY

The research questions that served as the basis for our experiment in a cursory look are stated below.

- What has the technological advancement contributed as regards personalised search systems?
- How do we assist people in finding exactly what they intend, even when they are unable to describe what they want in vivid terms?
- How best such a personalised search system would be realised?

To this end, in this section we formulated a model for the representation of the objects in our problem domain. Our problem domain consists of the web, social networks, citations, etc. Since most of these objects have inherent hierarchical structure, we defined a model that expressively takes advantage of this fact, while at the same time incorporating user's preferences or bias.

We first discussed a hierarchical model for social networks, and then generalized it to encompass all the objects in our problem domain. In social network analysis, social networks are commonly represented by directed or 'link' graphs. The vertices $i, j \in I$ represent the individuals in the network, while the relationships among them are depicted by the edges. The relationship between an individual $i$ and $j$ is denoted by the link from vertex $i$ to $j$. It is important to quantify the association among members of the network, to achieve this, nonnegative real-valued weights are assigned to each link. The weights can be interpreted as the level of 'endorsement', one member grants another. For instance, the weight on the edge $i \to j$ can be viewed as how much member $i$ endorses or respects member $j$; its value significantly indicates how $j$ is rated among its peers in the network. This abstractive illustration showed the importance of individuals' perception within a social media network. The aggregation of individual judgment is what bring about a finalistic rating within the network. Hence, formation of judgement graph can be achieved by processing link graph as in the case of a typical tree structure but in this case leaves are denoted with vertices. In the hierarchical model representation (see fig. 3.1) of the social network, the individuals are the leaves, while the intermediate levels are referred to as the interior nodes.

Our model easily generalized to all the objects in our problem domain. They can be represented as hierarchical social networks. For instance, in the Web, the web pages are the individuals, the hyperlinks represent the network while hierarchy represents the domains and sub-domains. This representation can be carried in diverse form depending on the subject domain. It is applicable I any knowledge representation and denotations. For instance, in a citation index, publications can be referred to as individuals, the references from one publication to another can be denoted as the network structure while the classification by subjects and sub-subjects defined an explicit hierarchical structure.

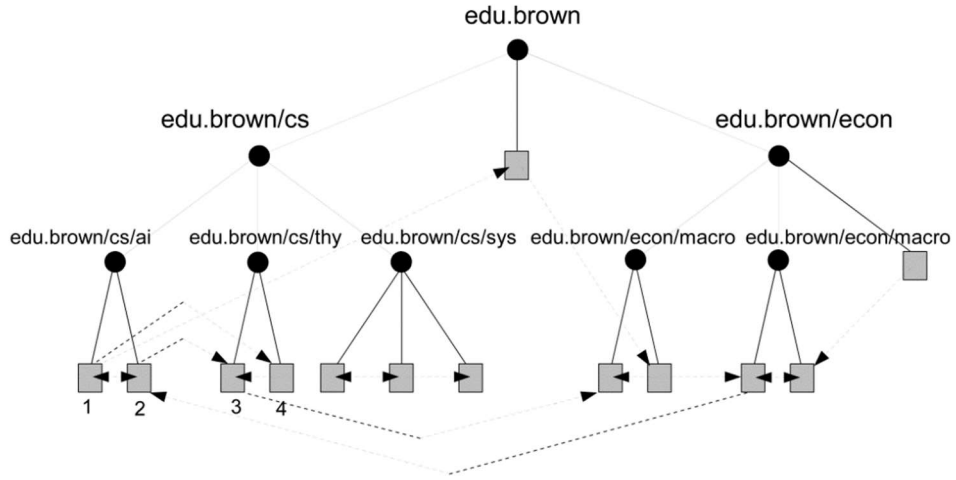


**Fig..1. Hierarchical Representation of a Web domain**[1]

[1] From Greenwald and Wicks (2006)

We incorporated user's bias into the hierarchical structure by introducing user's criteria at the leaves of the tree (solid rectangles). These criteria are the qualities identified by users as important; they form the basis for ratings and ranking. A sample personalised hierarchical model is shown in fig. 3.2—a network of the faculties and departments within a school hierarchy. At the top of the tree hierarchy are schools followed by faculties and departments to form a complete tree structure. The departments, indicated by solid circles, are the individuals in this society. The link relationship within the school hierarchy is represented by solid lines.

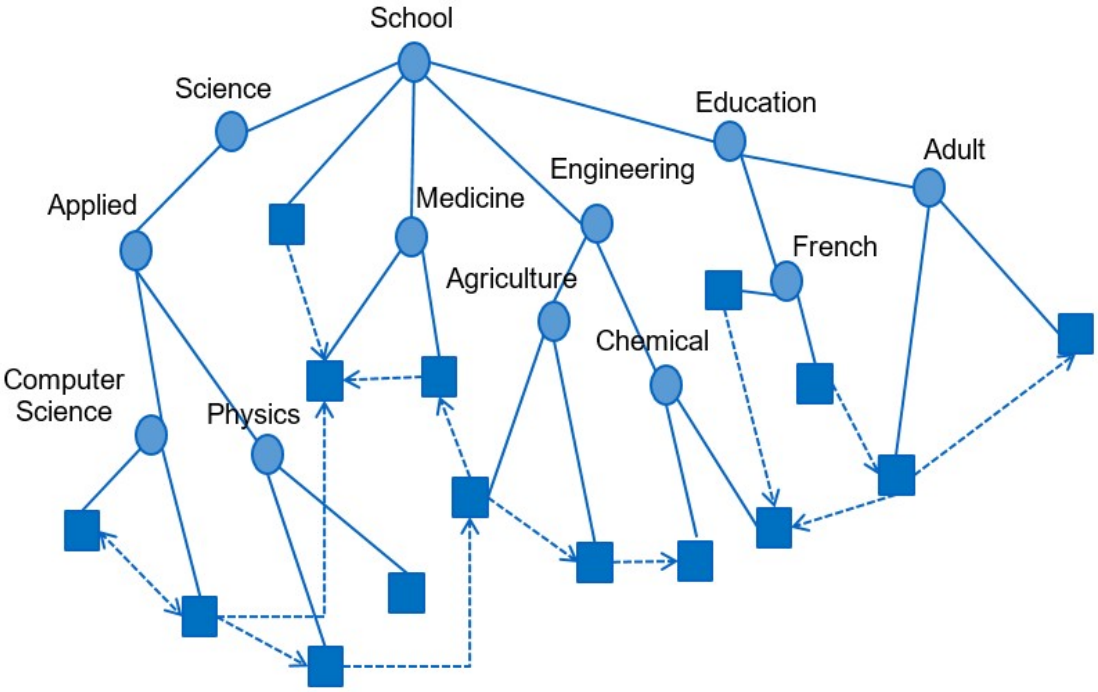


**Fig. .1. A Sample Hierarchical Representation of Our Proposed Model**

The relationships between criteria are indicated by dashed lines with arrowheads. Since individuals are expected to represent what they want in the best way it makes meaning to them, such a representation varies from user to user; hence, the hierarchical representation of the user domain together with the user-defined criteria could be modeled in the form of variant concept maps. Implicitly, various decision questions such as 'is-important', were included in the model to enable us arrived in a destination that closely matches the individual's expectation, based on the selected criteria. We referred to this sample hierarchical model in the course of the formulation and analysis of our algorithm. We noted that the general principles of our approach can be extended to any kind of network, which can be represented by a hierarchical structure.

Modelling a hierarchical representation structure like we have in the school example above underlines the importance of every individual with the network structure. The strength of each individual cumulatively form the overall rating of each object within the hierarchy. To have a good school, all the element within school structure must be takin into consideration. In addition, the individual has varying interest while making decision regarding the all factors and elements within school ecosystem. Differential in preferences would definitely lead to different ratings and choices.

Having formulated the personalized hierarchical representation of our search object, we proceeded to develop a recursive ranking algorithm that explicitly took advantage of this structure.

## 2. ENHANCED RECURSIVE RANKING ALGORITHM

In the preceding section, we described a model for a hierarchical social network that incorporates user's bias into the structure. He we discuss the process of designing and development of the algorithm. The algorithm is designed to exploit the underlying hierarchical topology of the network as presented in the previous section. The rating and ranking were personalised using user-defined preferences as biases. In order to ensure that our algorithm adequately solves our research problems, we proposed requirements, which it must satisfy:

1. Hierarchical structure of the network. The algorithm must be designed to take advantage of the inherent hierarchical structure of social network. This implies ease of transverse through the hierarchy of trees, in a Tree, Node, and Level (TNL) manner.
2. Personalization of ranking. The algorithm must reflect and address the issue of individual preferences by implementing a ranking procedure that incorporates users' biases within the network structure.
3. Resistance to link-spamming. Link-spamming is refers to exercises that are carried out with the aim of altering the ranking of a node, by creating many false nodes to that node. The false nodes are also called sybils (Cheng and Friedman 2006). A common example of link spanning is Web spamming (Gyongyi and Garcia-Molina 2004). Link-spamming presents a serious problem in ranking systems, especially in large networks. It reduces the efficiency of the ranking algorithm. A good ranking algorithm should demonstrate remarkable resistance to link-spamming.
4. Scalability. Problem domain that are represented in link-tree hierarchical structure are usually complex in nature. As a result, computations can be resource-intensive as the problem size increases. The algorithm must be scalable to handle the dynamic growth of the problem representation. Recursion presents the best approach. It would make the proposed solution more scalable, for efficient handling of future growth within the network. With recursion, the algorithm could be broken down into modules which are developed into components and then recursively applied across the domain.

Among the existing ranking algorithms reviewed in the previous chapter, the most similar to our proposed algorithm is QuickRank algorithm, which served as the prototype. We modified the QuickRank algorithm to incorporate user's bias. This way, we retained the desirable advantages of the algorithm, while providing adequate solution to the research problems. The selection of QuickRank was based on its desirable characteristics:

- QuickRank was designed with ease of transverse through the hierarchy in a Tree, Node and Leaf (TNL) manner. Hence, it is very suitable for developing solution to problems that can be represented in TNL fashion. It is able to take advantage of the inherent hierarchical structure of the social network resolving accordingly from leave to tree and vice versa as the case may be.
- QuickRank is recursive in nature, offering scalability, ease of parallelization, and ability to update. With this, problems can be broken down into modules with each module as a unit serving as input into another module or the main.
- QuickRank has proven to be remarkably resistant to link-spamming, and efficiently handles future growth within the social network.

Using QuickRank as prototype, we introduced a new ranking algorithm called "*Enhanced Recursive Ranking Algorithm (ERRA)*". Our solution differed from QuickRank by the introduction of a third parameter to the search—user bias. We now discussed the algorithm in details. In the course of the discussion, differences between QuickRank and the ERRA were pointed out, wherever such difference exists.

### 2.1 Explanations of the Techniques Adopted.

We defined concepts, which were essential for the understanding of the algorithms. These are major principles and techniques that were adopted in the course of developing our algorithm. Understanding of these concepts and terminology as applies to our research work is key.

#### 2.1.1 Social network

Social network can be described as a society in the form of an ecosystem that comprises of individuals which relates among one another. This kind of network can be represented in various forms depending on the nature of the problem. In the case of our research, we prefer representation of social network in the form of a link graph using graph theory. Mathematically, individuals and their relationships within the network can be represented for example as $i, j, k \in I$ . With this notaion, different relationship patterns can be enumerated from the expression. Such relationship can be converted to weights which serves as a measure for determining the strength of each of those relationships. The strength are then converted into a non-negative real values for computational and numerical analysis. With such representation, one can draw different inferences from various relationships and strength aggregation can be carried out so as to derive the value for resultant strength.

#### 2.1.2 Judgment

This refers to the ratings of the ranking of the individuals with the network. The underline principle is that every member of the network does not necessarily possess the same ranking weight. The merit of individuals within the society is a major factor in determining the weight of their ranking. Judgement is simulated into a mathematical expression to generate a non-negative real-value vector with individuals as indices. In our notation, we represent individuals as *I* and relations of individual judgement as *r*. The equivalence of relations between any two individual can be represented as $cr_1 = r_2$ where $r_1$ and $r_2$ represents the judgement relations for each of the two individuals. Ranking is represented here based on the relations for every individual judgements within the network and as an equivalence class $< r_i >$. Based on probability distribution, every individual ranking has one representation. Judgement relations for every individual within the network can be expressed consistently and it provides an estimate of the relative merits among the pairs of individuals involved. For instance, considering pairs of individuals i and *j*, the ranking of *i* relative to *j* can be expressed as $\frac{r_i}{r_j} \in [0;\infty)$ . This can be defined for every ranking in the network.

#### 2.1.3 Link graph

Link graph can described as a graph that denotes a network and its associated connectivity in a symbolic form. It is usually comprises of nodes and links that connect those nodes. The nodes represent the individuals while the link represent the relationship among the individuals. As described in previous section, this links can be simulated into weights that provide a means of evaluating the strength of the relationship. Due to the complexity of the problem domain, the weight here are represented as non-negative, real-value square matrix. The index on the matrix is on the individuals in the network represented as *I*. These weights can also be viewed as endorsement among the individuals in the network rather than a mere distance between to objects.

Higher weight from node A to B means individual A has high level of endorsement for individual B but not vice versa. A modified form of link graph which possesses positive diagonal entry values is referred to as judgement graph. In such graph, individual judgement is represented by columns.

### 2.1.4 BaseRank procedure

BaseRank is a ranking mechanism that computes and aggregates all the ranking parameters for the elements in the hierarchical structure that was described in previous section. It forms the core of our ranking algorithm by precomputing values for all the objects in our problem domain, we can then apply users' bias on the precomputed value to achieve the desirable results. Some of the BaseRank procedures that were evaluated in the course of this research are in-degree, out-degree, eigenvector centrality and PageRank (Greenwald and Wicks 2006). The output from BaseRank procedure is referred to as posterior ranking result and is a key input in the development of a recursive ranking algorithm. This output from BaseRank is then normalized into a probability distribution for easy integration. With BaseRank procedure, all the judgements of individual within a network can be aggregated and computed to generate only one aggregated posterior ranking. For illustration, assuming there is a graph R and a prior ranking $< r >$, by applying Bonacich's hypothesis, an inference to a collective judgement can be expressed as $r' = Rr$.

### 2.1.5 Parameterisation

This refers to a process of formulation and describing the parameters that are used with a model or system. It is a major requirement in the defining the specification and characterization of a model. The end product of parameterisation is usually in the form of expression or equation which can be applied as part of the components of the model. For instance, QuickRank is parameterised by BaseRank procedure.

### 2.1.6 Bonacich's hypothesis

Bonacich's hypothesis deals with the how to rate relative importance of each individual in a network. It does not just consider the ratings of individuals by its peers but also the inherent rating of the peers themselves. Consequently, the importance of an individual in a network is determined by the importance of those that endorsed him/her. It further means that the strength of an individual endorsement is directly proportional to the ranking of the individuals in the network that are making the endorsement. In other words, the collective ranking of an individual in a network is the total sum of individual judgments with the weight determines by the implicit ranking. Bonacich's metric, based on this hypothesis, is also known as eigenvector centrality.

### 2.1.7 Peer-review principle

(Greenwald and Wicks 2006) Peer-review principle states that endorsements among peers of individuals on the same sub-tree or community is more appropriate and should be taking seriously. Individuals at the same level in a network has many interaction and relationship and therefore rating in such situation should be fairer. In peer-review every other endorsement can be seen as approximation as the individuals may not possess enough information to make the right judgement while rating other individuals outside their community. As a result, while designing ranking algorithm, more weight should be assigned to ratings by individuals in the same community than others. At the heart of the QuickRank algorithm are **Bonacich's hypothesis** and "**peer-review principle**".

### 3.2.2 Ranking

Ranking is a process that is applied to resolve the complexity in searching experience. Due to huge amount of information and information sources on the web, a typical search query normally returns thousands of records. This presents a herculean task for a user trying to go through these thousands of information sources get the right or most suitable information. Bearing in mind that such users usual have a limited time. It is therefore imperative to provide a mechanism which will enable users to see the most relevant information first to enable them make quick decision. The developers of search engines then built ranking mechanism to solve this problem and make searches more efficient. With ranking algorithm, when a user issue a query to a search engine, information are displayed in order of relevance to the query parameters. This way users can check through few links and not bother about the rests of the information.

Technically, when a user issue a query, search engines behalf in form of a meta-search in which case, the query is send to many sub-sections of the search engines. The data resulting from the query is ranked and presented in form of a single list with the most relevant information at the top of the list. Given a query, a typical Meta search engine sends it to several search engines ranks the retrieved data and merges them in a single list. The ranking is carried out using two different techniques. The first method is based on the assumption that the weight allocated to the retrieved data initially by the search engines are known. The second method ignore information about the prior weights.

As researches have shown that the search engines are not necessarily using the same scale in calculating the similarity scores, it is therefore necessary to normalise the similarity scores so as to be able to compare the results from different search engines. In addition to this, the reliability of every search engine within the meta-search engines ecosystem must be measured and integrated into the ranking algorithm. One good way to achieve this is to assign weight factor to every search. The more relevant search information is return, the higher the weight score for the search engine.

Therefore, normalization is required to achieve a common measure of comparison. Moreover, the reliability of each search engine must be incorporated in the ranking algorithm through a weight factor. This factor is calculated separately during each search. Search engines that return more searched information should receive higher weight. This is due of the perception that the number of relevant data retrieved is proportional to the total number of data retrieved as relevant for all search engines exploited by the Meta search engine. [Dumais, 1994, Gavano, 1998, Towell et. al, 1995] raised the issue of compatibility and comparability of the weight scores among search engines despite normalisation. This was further proved by the fact that the results from search engines are often different even when the same query is applied. Based on [Y. Rasolofo et. al, 2001], conclusion could be draw that the collection used in search engines play a major role in influencing the scores. [Gavano, 1998] also argue and doubt the feasibility of comparison among the search engines. He then proposed that the results of the scores is not enough without providing the statistical analysis to back such result.

Other ranking algorithms were proposed [Aslam et. al, 2001] which do not considered the scored allocated to the retrieved data. Example of such ranking algorithm is Bayes-fuse which made use of probabilistic theory in determining the relevance of results to the query issued by the user. Another example is Borda-fuse which in concerned with ranking algorithm that is built based on the underlining principle of democratic voting. With Borda-fuse, the search engines return results inform of voting whereby the Meta search aggregates all the votes and ranking is done by adding up all the votes in a democratic way.

## 3. THE DEVELOPMENT OF ENHANCED RECURSIVE RANKING ALGORITHM

As already stated, the personalized hierarchical model of the underlying network directly incorporates user bias. ERRA, working on the model, introduces the user bias as a real-valued weight at both the leaves and nodes levels of the tree. To illustrate the algorithm, we used tree analysis which consist of judgement graph $R$ that contains vertices which are represented as leaves of tree $T$. There is the assumption that the link graph is pre-computed to produce judgement graph. The ranking of the individuals in the network is then equivalent to aggregating the individual judgments into a single collective ranking. According to Greenwald and Wicks (2006), the global or resultant ranking within a social network hierarchy can best be computed by first computing the local ranking of all the nodes in the network, and then use chain rule to aggregate all the local. This follows the important fact, that all rankings are considered as a form of probability distribution before they are normalised. The ranking is done by firstly at the leaf level based on the link information that is obtained from the processing of marginal subgraphs. The local rankings are then propagated and aggregated across the network in other to derive the absolute global ranking. Nodes ranking are computed recursively, to determine a node ranking, one needs to aggregate the local ranking of the node and the local ranking of all the associated interior nodes and leaves. A BaseRank procedure is used to compute the local rank. Prior ranking of the leaves is taken as an input, and posterior distribution as an output.

Our algorithm is bottom-up, starting from the bottom of tree $T$, the algorithm starts by collapsing the most posterior node. Then it progress gradually by continuously identifying the next nodes and collapse accordingly. This is done progressively until there are no more nodes to collapse in which case, the tree $T$ becomes a leave node. Assuming there is a node $n$, the process of collapsing this node involve three major steps as highlighted below:

i. Computing a local ranking at $n$ (the rankings of $n$'s children); this including the computation of local ranking for all the nodes that are associated with $n$ both interior and posterior.
ii. Computing $n$'s ranking and judgement; this is achieved by aggregation of all the local rankings and judgements of $n$'s children respectively.
iii. Aggregating the bias for $n$'s children into only one weighted value which is an associate of $n$. This is where the ERRA differs from QuickRank—QuickRank does not include this step. User's bias $\beta$ is a pre-computed weighted value that is applied to all $n$'s children. The value usually ranges from 0 to 1 ($0 < \beta \leq 1$, where $\beta$ is the user's bias at leaves level). User's bias of 1 means the QuickRank value for the leaf or node stays the same. If the user's bias for all the leaves, interior and posterior nodes is equal to 1, then our Enhanced Recursive Ranking Algorithm returns the same ranking results as QuickRank. We require that the user bias be normalised (the reason and the method for bias normalisation are discussed in section 3.3).

It is easy to see that ERRA just like QuickRank is a well-defined algorithm. The global ranking does not depend on the order of computation of the local ranking. This is made possible because the algorithm adopted the approach of propagating local computations from bottom to the top of the tree while to generate resultant global values. ERRA is designed to be deterministic and terminate when there are no more nodes to collapse and the root node becomes a leaf node.

**Data Structures:** In ERRA, the two main input are; $T_n$ which is a subtree of $T$ at node $n$, and the user bias which defines user specific presence. The output is return in the form of data structure as enumerated below:

i. Local ranking of the leaf nodes that are in the tree up to the support point $T_n$
ii. Computed judgement, that is the mean of all judgements of $T_n$'s leaves weighted by the ranking computed in (i).
iii. A user bias, computation of the mean value for biases of $T_n$'s leaves.

QuickRank returns only (i) and (ii).

At the leaf node $n$, the ranking follows a normal probability distribution with all weights on $n$ (denoted by $e_n$), the judgment is given by $R_n$ and bias as $\beta_n$.

**Computing Local Ranking:** Our algorithm relies on QuickRank's BaseRank procedure such as In-degree, Out-degree, Eigenvector centrality or PageRank (see Greenwald and Wicks 2006). BaseRank procedure takes $n$'s local prior ranking and a local judgment graph $M$ as input. For $j$ and $k$ both children of node $n$, the entry of $M$ in the row corresponding to $k$ and the column corresponding to $j$ is the aggregating of all endorsements from leaves in $T_j$ to leaves in $T_k$. This entry is equal to the sum of all entries in the $j$th judgment corresponding to leaves of $T_k$.

To facilitate recursive implementation of our algorithm, we decided to localise the process of computing the local ranking for all the nodes while ignoring the links to and from the sub-tree. Localizing of marginal rank computations is motivated by the aforementioned "peer-review principle". We adopted QuickRank's definition of approximate endorsements by individuals outside the sub community. For instance, considering an individual $i$ who is a member of community $A$ and another individual $j$ belonging to a different community $B$, An endorsement by $i$ in $A$ for $j$ in $B$ is as a component reflecting part of an endorsement by $A$ of $B$. Thus, endorsements by $A$ for individuals in $B$ are aggregated into an endorsement by $A$ of $B$ by first scaling the endorsements from each $i$ to each $j$ by $i's$ local rank, and then adding the resulting weighted endorsements. Replacing $j$ by any other $j'$ in $B$ does not change the aggregated endorsement. This is why the main endorsement is considered to be estimated. If the links starting at $i$ are interpreted to mean $i$'s judgment, then this collection procedure is considered to be based on Bonacich's hypothesis. That is, obtaining endorsements of $j \in B$ by $A$ is followed by aggregating over all $j \in B$ (to derive an endorsement of $A$).

**Aggregation of Rankings and Links:** Based on QuickRank, the ranking of $n$'s $m$ children into a single ranking corresponding to $m$ can be achieved by calculating the means of the rankings $r_1, \cdots, r_m$ according to the weights indicated by the local ranking $r$. Concatenating the $m$ rankings into a matrix $Q = \begin{bmatrix} r_1 & \cdots & r_m \end{bmatrix}$, the aggregation of ranking is simply expressed as $Qr$. Also corresponded with each child $j$ of a collapsible node $n$ are a judgment $l_j$, and a user bias $b_j$. These judgments and user bias are computed using the exact same techniques as the case of rankings.

Having described the major steps of our Enhanced Recursive Ranking Algorithm, we now present the algorithm in pseudo code (Table 3.1) (we refer the reader to Greenwald and Wicks (2006) for a detailed explanation on BaseRank procedure). In the pseudo code, weighted judgment is denoted as $e_n$, weighted bias as $\beta_n$. The matrices are denoted by $P = [l_1 \quad \cdots \quad l_m]$ - judgment; $Q = [r_1 \quad \cdots \quad r_m]$ - ranking; $B = [\beta_1 \quad \cdots \quad \beta_m]$ - bias.

**Table 1. Enhanced Recursive Ranking Algorithm**

| | *Algorithm: Enhanced Recursive Ranking (Collapsing node n)* |
|---|---|
| 1: | ***if*** *(n.isLeaf())* ***then*** |
| 2: | ***return*** *(({n.getJudgment(), en}), Concat({n.getUserBias (), βn}));* |
| 3: | ***Else*** |
| 4: | *m = n.numChildren ()* |
| 5: | ***for*** *(j = 1 to m)* ***do*** |
| 6: | $(l_j, r_j, b_j) \leftarrow$ *QuickRank(n.getChild(j))* |
| 7: | ***for*** *(k = 1 to m)* ***do*** |
| 8: | $M_{kj}$ *= Sum* $(l_j$*, n.getChild(k))* |
| 9: | ***end for*** |
| 10: | ***end for*** |
| 11: | $P = [l_1 \quad \cdots \quad l_m]$ |
| 12: | $Q = [r_1 \quad \cdots \quad r_m]$ |
| 13: | $B = [\beta_1 \quad \cdots \quad \beta_m]$ |
| 14: | *r = BaseRank(M, n.getLocalPriorRanking())* |
| 15: | ***return*** *(*$(P_r, Q_r, B_r)$*);* |
| 16: | ***end if*** |

In the sample school structure described in previous section, using ERRA to compute the ranking of any department or faculty, we must compute the marginal (local) rankings of interior and posterior nodes (faculties and departments) and compute the cumulative local rankings using chain rule. Ranking techniques will be applied on each individual within the hierarchy. We begin by restricting the link graph a particular faculty, thereby constructing a local link subgraph including only the departments in that faculty. Then using BaseRank procedure, we compute the marginal ranking of the departments. Then we scale the links from this faculty's departments to the outside departments by the marginal ranks of the faculty's departments. Finally, the results are added to generate a resultant link which is represented from the current faculty to the outside connected departments. Furthermore, we progressed to recursively construct a marginal link sub graph, and generated corresponding ranking values for the faculties. The combination of the two local ranking for the departments in the faculty node produced one marginal ranking for all the departments in the faculty. Repeating this procedure for the other faculties, we obtain a single global ranking for each faculty. This process is repeated till the hierarchy is fully collapsed into a single a node.at this point, the ranking of all the departments in the school are generated.

Thus, obtained rank is the same as the QuickRank of the departments. We then scale the QuickRank using the pre-computed user bias aggregated at each node, to obtain the enhanced recursive ranking. If the user bias in all the nodes is not equal to 1, then the final ranking may differ from the QuickRank.

## 4. NORMALIZATION OF RANKING CRITERIA

As stated in the description of the steps of the algorithm, it is required that the user bias be normalised. We explain why normalized bias value is needed, and the method of normalization. The third parameter in our algorithm, user bias, incorporates individualism into the ranking. The values of the user bias are computed from the user-defined search criteria. The order of preference of the search criteria (which may be specified explicitly by the user, or inferred implicitly from their profile) is important in the ranking of the results of the search. It is more convenient to allow users express these criteria in a form, most familiar to them. For instance, they may specify them as being 'very important', 'important', 'not important', etc. However, the quantification of these terms is arbitrary, and may not reflect the reality of the user's intentions, as different users may understand these terms in different ways.

Another possible scheme is to allow them specify them on a scale of importance, for example, 0-10, 1-5, etc. Most surveys, with which people are accustomed, usually use such schemes. It makes sense, therefore, to adopt such an approach to specifying user-defined preferences. In order to make a direct comparison among criteria on their scale of relevance, it may be required that the cumulative sum of all the criteria equals a fixed number (possibly the maximum limit of the range). For instance, if we are using the 0-10 scale of relevance, it may be specified that the sum of the values, assigned by each user to the criteria must be equal to 10. Say, there are four criteria—newness, proximity, cheapness, ease of use. Then a user could assign any non-negative values to these criteria, as long as the sum equals 10. For example, one may assign 5 to newness, 3 to proximity, 0 to cheapness, 2 to ease of use; or 2.5 to all the criteria. We could then interpret the first example as the user considered newness to be of highest importance among the criteria., the next is proximity and then ease of use in that order, while the user does not consider cheapness important. In the second case, the user considers all the criteria to be equally important. Of course, while this requirement of constant cumulative sum enables us to quickly determine that order of preference of the criteria, as well as quantitatively determine the relative relevance of each, it is however, not very convenient for users to express their preferences that way. Users may know that some criteria are more important to them than others, but may not be able to quantitatively define how relatively important they are, that is, they are not able to say, whether criterion A is twice as important as criterion B. Moreover, such a scheme may not reflect the reality of user's preferences.

As stated in the introduction to this chapter, a key question in this research work is determining a way to help users find what they intend, when they cannot specify in vivid terms exactly what they want. It is important, therefore, to allow users express their preferences in the form most familiar and convenient to them. To this end, we shall adopt one of the methods used for surveys—using a scale of preference to specify user search criteria. The scales are determined individually for each criterion, in such a way that, they make the most sense to the users. To illustrate, using our previous examples with four ranking criteria, we may adopt a scale of 0-10 for newness and cheapness, 1-5 for proximity, and 0-100 for ease of use. Of course, the same scale could be used for all the criteria, if it makes sense. However, the adoption of individual scales for the criteria, while very convenient, poses a problem - how can we determine the order of preferences, since different scales are used? To solve this problem, there is need to bring the values to the same scale for ease of comparison. This is achieved through normalization of the values.

There are various normalisation techniques in use today—min-max, cosine, non-monotonic, Z-score, etc. In other to select the best normalisation method to use, there is need to consider the purpose of such normalisation, the statistical properties of the data, and so on. Z-score normalization is useful if the distribution of the data is normal, which is not the case (or cannot be proven to be the case) for the values assigned the criteria. Local linear rescaling using quantile or percentile values alleviates the so-called 'fat tail' problem (Ready and Hu 1995), and therefore works well if there are outliers in the dataset, which is not the case here. For our purpose, we shall use the linear min-max normalization method. Since the linear min-max normalization does not distort the relative pairwise distances (because it is linear), it is the most appropriate for making our various criteria comparable without introducing additional errors. In min-max normalization, care must be taken to check for the *min=max* case, which luckily in our case, we do not need to worry about. It is computationally easy too. The formula for the min-max normalization is given by:

$$newX = \frac{\max - X}{\max - \min} \quad (3.1)$$

Where $newX$ the normalized value of the criterion is, $X$ is the user-specified value of the criterion, *max* refers to maximum limit of the scale, *min* represents minimum limit of scale. The min-max method normalizes each of the criteria in the range $[0;1]$. We shall interpret the value of 0 as meaning that the criterion is not important, while 1 signifies that the criterion is essential to the search. Larger values of the criteria indicate higher importance.

Our algorithm uses the min-max metric to normalize the ranking criteria. The purpose of normalizing the criteria is to be able to compare the order of preference of the ranking criteria, specified by the user. It also allows us to compare search results across similar data from different datasets. Using the min-max metric ensures that additional errors are not introduced by the normalisation step, as the relative distances between values are preserved. This method allows us to determine the importance of each of the ranking criteria in regards to the relationship between the QuickRank of the objects and their respective position in enhanced recursive ranking. The idea behind this normalisation metric is that most criteria are as important as the size of their normalised values, which in turn ensures that highly relevant results possess higher ranks in the generated list.

Finally, it is important to note that normalisation of the criteria is a pre-processing step, just like converting the link graph into a judgment graph. The user bias incorporated into our algorithm is actually not the user-specified values, but instead the normalized values. In this work the values of all the criteria are being normalized by using a min-max factor of the user-defined criteria, which removes the difficulty of the comparison of the relative importance of the ranking criteria, and allows us to compare search results across different datasets.

## 5. IMPLEMENTATION OF ENHANCED RECURSIVE RANK ALGORITHM

One of the deliverables of this research work is a software program that demonstrates our ranking algorithm. Here, we present explanation about the tools and techniques that were deployed for the implementation, special characteristics and limitations of the program. The structure of the software implementation of our enhanced ranking algorithm is shown on fig. 3.3 below. The components are discussed briefly.

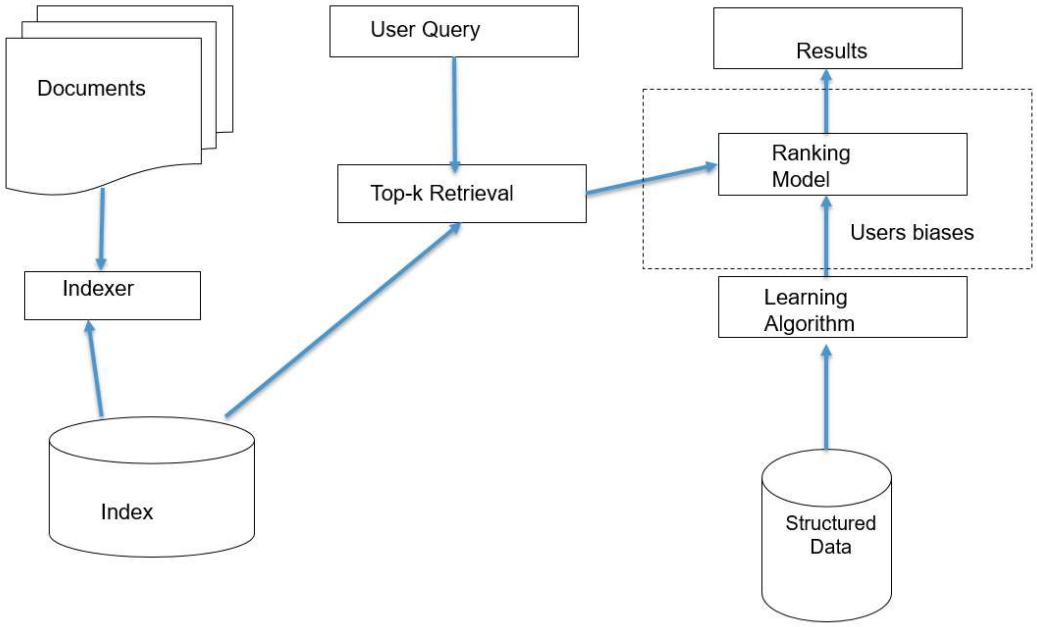


**Fig. 2. Enhanced Recursive Ranking Model Implementation**

i. **Indexer:** Our ranking algorithm enables users search for what they want, specifying the criteria and properties in order of preference. Of course, the traits will vary in importance to different users. A recursive search through a huge dataset places high demand on the processing power of the processing machine. It is therefore, required that the database be perfectly normalized and properly indexed. The task of indexing the records in the database is carried out by an indexer, inbuilt into the ranking system. Indexing makes the search easier, and ensures the fast operation of the database.

ii. **User Query Handler:** To enable users input data, select search criteria, traits, qualities, etc., a robust and user-friendly interface is needed. The interface is developed in accordance with industrial standards. It receives user's query and link it to most relevant profile on the network. A **query processor** passes the queries through a web mining process, and returns the most relevant results.

iii. **Database Manager:** A robust database is required to manage the huge volume of data. Owing to its ease of use, and free availability, MySQL database is used in the design of the ranking system.

iv. **Web crawler:** A web crawler is used for the selection and gathering of the standardized dataset, which is used for the testing of the algorithm. From the web, it gathers relevant data, which forms the foundation of the database. The selected information will target people and their various unique characteristics, and will be stored in the database for referencing. The crawler is designed with the ability for data gathering from online social networks through their Application Programming

Interface (API). Using the APIs, it is possible to extract relevant information from the existing social network database, including data about the structure of the network. A popular example is Facebook API).

v. **Ranker (or Ranking Model):** The ranker measures the importance of the search results returned, using the proposed Enhanced Recursive Ranking Algorithm. The ranker incorporates user bias in computing the ranks of the results.

vi. **Retrieval Engine (Top-k Retrieval):** It performs lookups on the index tables against the search query.

vii. **Learning Algorithm**: This implements a machine learning algorithm, which helps the system to adapt itself over time to rank results better, by analyzing previous data.

i. Our program is implemented using the Python and JavaScript computing language. The selection of these languages is based on the ease of programming, availability of computational libraries, as well as toolkits for GUI programming.

## *5.* THE RANKING DISTRIBUTION

As stated previously, the ranking output by the algorithm is a probability distribution. From the statistical properties of this probability distribution, useful information about the dataset, and consequently the underlying hierarchical social network, can be extracted. Two of the most important properties are the **weighted mean** and **variance** of the distribution. **Weighted mean** enables us to calculate the estimated value of the rank given user bias. This is calculated by

$$\overline{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}{w_1 + w_2 + \cdots + w_n}$$

Where $\{x_1, x_2, \cdots, x_n\}$ is the dataset (ranks), $\{w_1, w_2, \cdots, w_n\}$ are the non-negative weights of the criteria. The weights are assigned according to the importance of the criteria at each node. It can be seen that criteria with a high weight contribute more to the weighted mean than do criteria with a low weight. We normalize the weights, such that their sum equals 1, that is $\sum_{i=1}^{n} w_i = 1$. Consequently, the formula above gets simplified to

$$\overline{x} = \sum_{i=1}^{n} w_i x_i = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n .$$

If equal normalised weights are assigned to all the criteria, then we get

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

On the other hand, **variance** shows the 'spread' of the ranks. Higher variance signifies a wide difference in the relevance of the returned results, while lower variance means that the results are more or less on the same level of relevance. The variance of the distribution is computed by:

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Where $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the expected value.

The weighted mean and variance are very significant, as they help us compare the degree of similarity between two standardized datasets (rankings) of the same network.

It is possible to determine the approximate **probability distribution of the rank**. In statistics there are numerous techniques (Ritzema 1994) for probability distribution fitting to a series of data, including parametric methods (Cramér 1946) and regression method. The parametric methods include; method of moments, method of L-moments (*Hosking 1990)*, maximum likelihood method *(Aldrich 1997).* In this work, we use MATLAB's ALLFITDIST function, which fits all valid parametric probability distributions to data (MATLAB 2016).

Note that for non-parametric kernel-smoothing, FITDIST should be used directly instead. The probability distribution enables us to determine the dependence of the rank on different factors, from which useful information can be inferred.

The **correlation** between the original ranking (using QuickRank, or another ranking algorithm) and the individually biased ranking can be established. This correlation enables us to determine the influence of user bias in the ranking. This way, we could determine the efficiency of our algorithm in 'personalizing' search results based on user-defined preferences. Correlation can be computed using multiple methods. In this work, we adopt the Pearson Correlation Coefficient. The choice of method is based on the ease of computation, as well as the structure of the underlying test datasets. The Pearson Correlation Coefficient is given by:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sigma_X \sigma_Y} = \frac{E[XY] - E[X]E[Y]}{\sqrt{E\left[X^2\right] - \left(E[X]\right)^2}\sqrt{E\left[Y^2\right] - \left(E[Y]\right)^2}}$$

Where *cov(X,Y)* is the covariance, $\sigma_X$ is the standard deviation of *X*, $\sigma_Y$ is the standard deviation of *Y*, $\mu_X$ is the mean of *X*, $\mu_Y$ is the mean of *Y*, *E* is the expectation.

When applied to a sample, the Pearson Correlation Coefficient is given by:

$$r = r_{xy} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}} \quad (3.7)$$

for two datasets $\{x_1, \ldots, x_n\}$, and $\{y_1, \ldots, y_n\}$. To get the weighted correlation coefficient, we use the formula

$$\operatorname{corr}(x,y;w)=\frac{\operatorname{cov}(x,y;w)}{\sqrt{\operatorname{cov}(x,x;w)\operatorname{cov}(y,y;w)}} \quad (3.8)$$

where $\operatorname{cov}(x,y;w)=\frac{\sum_{i=1}^{n} w_i\left(x_i-m(x;w)\right)\left(y_i-m(y;w)\right)}{\sum_{i=1}^{n} w_i}$ is the weighted covariance; $m(x;w)=\frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$ is the weighted mean.

In our case we apply formula (3.7) to the rankings obtained with ERRA, and the original rankings obtained through QuickRank or another algorithm. The value of $r$ determines how strongly the introduction of user bias distorts the original ranking, how much 'personalized' by user criteria it becomes. Strong correlation coefficient can be interpreted to mean that user bias greatly influence the ranking result.

Another useful property is the **probability distribution of the search criteria**. It reflects how each criterion varies among users. The distribution shades light on the properties of the users, and could be used to train the system to predict and rank results better, using information inferred from the user's profile. For this purpose, a learning algorithm unit is included in the program (see fig. 3.2). To determine the probability distribution of each of the criteria, we use the ALLFITDIST function from MATLAB as used above.

### 5.1. Generality and Limitations of the Enhanced Recursive Rank Algorithm

Having outlined in details the workings, implementation and the properties of the ERRA, it is necessary that we discuss the generality and limitations of the algorithm. As previously mentioned, the general principles of the proposed approach can be applied to any social network with a hierarchical structure. This implies that our algorithm has a wide area of applications, since a lot of search objects can be represented with a hierarchical structure. From this fact, however, springs a serious limitation – the algorithm cannot be used directly with networks, which are not hierarchical. However, it is possible to modify the algorithm to work on such non-hierarchical networks, or modify the structure of the network to suit the algorithm. We have, however, not yet investigated this possibility. Despite this limitation, the algorithm can be applied to existing search engines as a means of personalizing search results.

## 6. CONCLUSION

We have discussed about the proposition of a new algorithm for hierarchical social networks. The new algorithm named "Enhanced Recursive Ranking Algorithm" (ERRA) is a modification of the QuickRank algorithm to incorporate user bias. The theoretical framework for the implementation of the algorithm was discussed, as well as the statistical properties of the results. From the theoretical analysis, we demonstrated that the algorithm shows lots of promises in solving the research problems proposed. In the next chapter, experimental study of the algorithm using the software program developed will be carried out. This will help us determine the adequacy of the proposed solution.

BIBLIOGRAPHY

1. Ankur, G & Rajni, J (2008). An overview of ranking algorithms for search engines. Proceedings of the 2nd National Conference; INDIACom-2008, New Delhi. Feb 08 – 09.
2. Arasu, A; Novak, J; Tomkins, A; Tomlin, J (2002). PageRank computation and the structure of the web: Experiments and algorithms. Proceedings of the Eleventh International World Wide Web Conference, Poster Track. Brisbane, Australia. 107-117.
3. Baeza-Yates, R; Saint-Jean, F & Castillo, C (2002). Web Dynamics, structure and page ranking. Proceedings of 9th International Symposium on String Processing and Information Retrieval, SPIRE, Springer LNCS, Lisbon, Portugal, 117–130.
4. Bates, M, J; Wilde, D, N, & Siegfried, S (1993). An analysis of search terminology used by humanities scholars: The Getty Online Searching Project report no. 1. Library Quarterly, 63(1), 1-39
5. Bates, M. J (1979). Information search tactics. Journal of the American Society for Information Science, 30(4), 205-214.
6. Bates, M; Idea, J. (1979). Tactics. Journal of the American Society for Information Science, 30 (5), 280-289.
7. Beel, J; Gipp, B; Stiller, J (2009). Information retrieval on mind maps – what could it be good for? Proceedings of the 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'09). Washington, DC: IEEE.
8. Belkin, N, J & Croft, W, B (1987). "Retrieval Techniques," in Annual Review of Information Science and Technology, ed. M. Williams. New York: Elsevier Science Publishers, 109-145.
9. Belkin, N. J (1996). Intelligent information retrieval: whose intelligence? In Proceedings of the 5th International Symposium for Information Science (ISI '96): Humboldt-Universität zu Berlin, 17. - 19. Oktober 1996; Krause, J., Herfurth, M., Marx, J., Eds.; Universitätsverlag Konstanz: Konstanz, Germany, 25-31.
10. Belkin, N. J. (1993). Interaction with texts: Information retrieval as information seeking behaviour. In Information Retrieval '93: Von der Modellierung zur Anwendung, Knorz, G., Krause, J., Womser-Hacker, C. Eds.; Universitaetsverlag Konstanz: Konstanz, Germany, 1993; 55-66.
11. Belkin, N. J.; Cool, C.; Stein, A.; Thiel, U (1995). Cases, scripts and information seeking strategies: On the design of interactive information retrieval systems. Expert Systems with Applications, 9 (3), 379-395.
12. Belkin, N. J; Marchetti, P, G; Cool, C (1993). Design of an interface to support user interaction in information retrieval. Information Processing and Management, 29 (3), 325-344.
13. Bellardo, T (1985). What do we really know about online searchers? Online Review, 9 (3), 223-239.
14. Bhavnani, S. K (2002). Important cognitive components of domain-specific search knowledge. In The Tenth Text REtrieval Conference, TREC-2001; Voorhees, E.M.; Harman, D.K. Eds.; Information Today: Medford, NJ, 571-578.
15. Bilal, D (2002). Perspectives on children's navigation of the World Wide Web: Does the type of search task make a difference. Online Information Review, 26 (2), 108-177.
16. Bjørklund, T. A., Götz, M., Gehrke, J., & Grimsmo, N. (2011, October). Workload-aware indexing for keyword search in social networks. In Proceedings of the 20th ACM international conference on Information and knowledge management (pp. 535-544). ACM
17. Bonacich, P (1972). Factoring and weighting approaches to status scores and clique detection. Journal of Mathematical Sociology, pages 113–120

18. Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software, 80(4), 571-583.
19. Broder A. Z., Kumar R., Maghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A. & Wiener J.L. (2000). "Graph structure in the Web", Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, pp. 309-320.
20. Bruza, P. D; Dennis, S (1997). Query-reformulation on the Internet: Empirical data and the hyperindex search engine. In RIAO 97: Conference proceedings with prototype and operational systems demonstrations: Computer-assisted information searching on Internet, McGill University, Montreal, Quebec, Canada, 25th-27th June 1997; RIAO 97, Ed.; CID: Paris, 1997; Vol. 1, 488-499.
21. Bryman, A. (2012). Social research methods. OUP Oxford.
22. Byström, K (2002). Information and information sources in tasks of varying complexity. Journal of the American Society for Information Science and Technology, 53 (7), 581-591.
23. Byström, K; Järvelin, K (1995). Task complexity affects information-seeking and use. Information Processing and Management, 31(2), 191-213.
24. Callahan, E. (2005). Interface design and culture. Annual Review of Information Science and Technology, 39, 257-310.
25. Chang, S (1995). Toward a multidimensional framework for understanding browsing. Unpublished doctoral dissertation, Rutgers University: New Brunswick, N, J.
26. Chen, H & Dhar, V (1991). Cognitive processes as a basis for intelligent retrieval system design. Information Processing and Management, 27 (5), 405-432.
27. Cheng, A & Friedman, E (2006). Manipulability of PageRank under sybil strategies. In First Workshop on the Economics of Networked Systems (NetEcon06), URL http://www.cs.duke.edu/nicl/netecon06/papers/ne06-sybil.pdf.
28. Cheng, Y., Park, J., & Sandhu, R. (2012). A user-to-user relationship-based access control model for online social networks. Data and Applications Security and Privacy XXVI, 8-24.
29. Chu, H (2003). Information Representation and Retrieval in the Digital Age; Information Today: Medford, N. J
30. Cole, C (2001). Intelligent information retrieval: Part IV. Testing the timing of two information retrieval devices in a naturalistic setting. Information Processing and Management, 37 (1), 163-182.
31. Craswell, N., Hawking, D & Robertson, S. (2001). Effective site finding employing link anchor information. In Proceedings of the ACM Conference on Information Retrieval (SIGIR '01), 250-257.
32. Dalal, M. (2007). Personalized social & real-time collaborative search. In Proceedings of the International World Wide Web Conference (WWW '07), 1285-1286.
33. Dieste, O., Grimán, A., Juristo, N., & Saxena, H. (2011, September). Quantitative determination of the relationship between internal validity and bias in software engineering experiments: consequences for systematic literature reviews. In Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on (pp. 285-294). IEEE.
34. Drabenstott, K. M (2003). Do nondomain experts enlist the strategies of domain experts? Journal of the American Society for Information Science and Technology, 54 (9), 836-854.
35. Duhan, N., Sharma, A. K., & Bhatia, K. K. (2009, March). Page ranking algorithms: a survey. In Advance Computing Conference, 2009. IACC. IEEE International (pp. 1530-1537). IEEE.
36. Dumais, S, T; Belkin, N, J (2005). The TREC interactive tracks: Putting the user into search. In TREC: Experiment and Evaluation in Information Retrieval; Voorhees, E.M.; Harman, D.K., Eds.;

The MIT Press: Cambridge, MA, 123-152.
37. Earhart, S. (1986). The UNIX Programming Language, vol. 1. New York: Holt, Rinehart, and Winston.
38. Ellis, D & Haugan, M (1997). Modeling the information seeking patterns of engineers and research scientists in an industrial environment. Journal of Documentation, 53 (4), 384-403.
39. Ellis, D. A (1989). Behavioural approach to information retrieval system design. Journal of Documentation, 45 (3), 171-212.
40. Faloutsos, C. (1985). "Access Methods for Text," Computing Surveys, 17(1), 49-74.
41. Fenichel, C. H (1981). Online searching: Measures that discriminate among users with different types of experience. Journal of the American Society for Information Science, 32 (1), 23-32.
42. Fidel, R & Pejtersen, A, M (2004). From information behavior research to the design of information systems: The cognitive work analysis framework. Information Research, 10 (1). http://informationr.net/ir/10-1/paper210.html.
43. Fidel, R & Soergel, D (1983). Factors affecting online bibliographic retrieval: A conceptual framework for research. Journal of the American Society for Information Science, 34 (3), 163-180.
44. Fidel, R (1985). Moves in online searching. Online Review, 9 (1), 61-74
45. Foote, Jonathan (1999). "An overview of audio information retrieval". Multimedia Systems. Springer.
46. Ford, N, Wilson, T, Foster, D, Ellis, A, & Spink, A, D (2002). Information seeking and mediated searching. Part 4.Cognitive styles in information seeking. Journal of the American Society for Information Science and Technology, 53 (9), 728-735.
47. Ford, N; Miller, D, & Moss, N (2002). Web search strategies and retrieval effectiveness: An empirical study. Journal of Documentation, 58 (1), 30-48.
48. Frakes, W, B. (1992). Information retrieval data structures & algorithms. Prentice-Hall, Inc
49. Franceschet, M (2002). "PageRank: Standing on the shoulders of giants".
50. Fujimura K., Inoue R., and Sugisaki M. (2005). "EigenRumor Algorithm for Ranking Blogs". May 10-14, 2005, Chiba, Japan.
51. G. Jeh, G & Widom, J (2003). Scaling personalized web search. In Proceedings of the International World Wide Web Conference ('03), 271-279, 2003.
52. García-Crespo, Á, Colomo-Palacios, R., Gómez-Berbís, J. M., & García-Sánchez, F. (2010). SOLAR: social link advanced recommendation system. Future Generation Computer Systems, 26(3), 374-380
53. Ghafari, M., Saleh, M., & Ebrahimi, T. (2012). A federated search approach to facilitate systematic literature review in software engineering. International Journal of Engineering, 23(5), 12-36
54. Goodrum, Abby A. (2000). "Image Information Retrieval: An Overview of Current Research". Informing Science. 3 (2).
55. Google technology overview (2004) "http://www.google.com/intl/en/corporate/tech.html.
56. Google. Personalized search for everyone. http://googleblog.blogspot.com/2009/12/personalized-search-for-everyone.html. Last recoup on April, 16, 2010.
57. Greenwald, A., & Wicks, J. (2006). QuickRank: A recursive ranking algorithm. In Proc. of the 1st International Workshop on Computational Social Choice.
58. Greenwald, A., & Wicks, J. (2006). QuickRank: A recursive ranking algorithm. In Proc. of the 1st International Workshop on Computational Social Choice.
59. Gupte, M., Shankar, P., Li, J., Muthukrishnan, S., & Iftode, L. (2011, March). Finding hierarchy in directed online social networks. In Proceedings of the 20th international conference on World wide web (pp. 557-566). ACM.
60. Gyongyi Z. & Garcia-Molina H. Web spam taxonomy (2004). Technical report, Stanford University

Technical Report.

61. Haveliwala, T. (2002). "Topic-Sensitive PageRank", In Proceedings of the 11th World wide Web Conference.
62. Hawk, W. B.; Wang, P (1999). Users' interaction with the World Wide Web; Problems and problem solving. Proceedings of the 62nd ASIS Annual Meeting, 36, 256-270.
63. Hema, D, B. & Roy, N. (2011). An improved Page Rank Algorithm based on Optimized Normalization Technique. International Journal of Computer Science and Information Technologies, 2 (5), 2183-2188
64. Hotho, A; Jäschke, R; Schmitz, C, & Stumme, G. (2006). Information retrieval in folksonomies: Search and ranking. In Lecture Notes in Computer Science. Vol. 4011/2006(the Semantic Web: Research and Applications), 411-426
65. Howard, H (1982). Measures that discriminate among online users with different training and experience. Online Review, 6 (4), 315-326.
66. Hsieh-Yee, I (1993). Effects of search experience and subject knowledge on the search tactics of novice and experienced searchers. Journal of the American Society for Information Science, 44 (3), 161-174.
67. Hyldegard, J (2006). Collaborative information behaviour: Exploring Kuhlthau's information search process model in a group-based educational setting. Information Processing and Management, 42 (1), 276-298.
68. Ingwersen, P & Järvelin, K (2005). The Turn: Integration of Information Seeking and Retrieval in Context; Springer. Germany: Heidelberg
69. Ingwersen, P (1992). Information retrieval interaction; Taylor Graham: London.
70. Ingwersen, P (1996). Cognitive perspectives of information retrieval interaction: Elements of a cognitive IR theory. Journal of Documentation, 52 (1), 3-50.
71. J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu & Z. Chen. (2005). Cubesvd: A novel approach to personalized web search. In Proceedings of the International World Wide Web Conference (WWW'05), pp. 382-390.
72. Jamshidi, P., Ghafari, M., Aakash, A., & Pahl, C. (2012). A protocol for systematic literature review on Architecture-Centric Software Evolution Research. Technical Report, Lero-The Irish Software Engineering Research Centre, Dublin City University.
73. Jansen, B. J. & Rieh, S. (2010). The Seventeen Theoretical Constructs of Information Searching and Information Retrieval. Journal of the American Society for Information Sciences and Technology. 61(8), 1517-1534.
74. Jansen, B. J.; Spink, A.; Saracevic, T (2000). Real life, real users, and real needs: A study and analysis of user queries on the Web. Information Processing and Management 2000, 36 (2), 207-227.
75. Joachims, T. (2002). Optimizing search engines employing clickthrough data. In Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (SIGKDD'02), 133-142
76. Jones, K. S., Walker, S & Robertson, S. E (2000). A probabilistic model of information retrieval: Development and comparative experiments. Information Processing and Management. 36 (6), 779-808, 2000.
77. Jones, S.; Cunningham, S.J.; McNab, R.; Boddie, S (2000). Human-computer interaction for digital libraries: A transaction log analysis of a digital library. International Journal on Digital Libraries , 3 (2), 152-169.
78. K. Bharat and G.A. Mihaila (2002). When experts agree: Employing Non-Affiliated Experts to Rank Popular Topics. ACM Transactions on Information Systems, 20(1), 47-58,.

79. Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering–a tertiary study. Information and Software Technology, 52(8), 792-805.
80. Kleinberg J. M. (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5), 604 –632.
81. Kracker, J (2002). Research anxiety and students' perceptions of research: An experiment: Part 1. Effect of teaching Kuhlthau's ISP model. Journal of the American Society for Information Science and Technology, 53 (4), 282-294.
82. Kuhlthau, C. C. (1991). Inside the search process: Information seeking from the user's perspective. Journal of the American Society for Information Science 1991, 42 (5), 361-371.
83. Kumar, R., Novak, J., & Tomkins, A. (2010). Structure and evolution of online social networks. Link Mining: Models, Algorithms, and Applications, 337-357.
84. L. Srour, A Kayssi and A. Chebab (2007). Personalized Web Page Ranking Employing Trust and Similarity", IEEE 2007.
85. Lau, T.; Horvitz, E (1999). Patterns of search: Analyzing and modeling Web query refinement. In Proceedings of the 7th International Conference on User Modeling Banff, Canada, June 1999; Kay, J., Ed.; Springer-Wien: New York, 119-128.
86. Lazonder, A. W.; Biemans, H. J. A.; Wopereis, I. G. J. H (2000). Differences between novice and experienced users in searching information on the World Wide Web. Journal of the American Society for Information Science, 51 (6), 576-581.
87. Lempel R. & Moran S. (2000). The stochastic approach for link-structure analysis (SALSA) and the TKC effect. Proceedings of the 9th. International World Wide Web Conference, Amsterdam, The Netherlands, pp. 387 – 401
88. Liu N. C. and Cheng Y. (2005). The academic ranking of world universities. Higher Education in Europe, 30(2), 1-14
89. M. Montaner, B. López, & J. L. de La Rosa (2003). A taxonomy of recommender agents on the internet. Artificial Intelligence Review. 19(4), 285-330.
90. M. Richardson and P. Domingos (2002). The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank", Advances in Neural Information Processing Systems, 14, 1441-1448, MIT Press.
91. M.S. Aktas, M.A. Nacar, and F. Menczer (2004). Personalizing PageRank Based on Domain Profiles, WebKDD.
92. Manning, C., Raghavan, P & Schutze, H (2008). Introduction to information retrieval. Cambridge: University Press.
93. Marchionini, G (1995). Information-Seeking in electronic environments; Cambridge University Press: Cambridge,.
94. Marchionini, G.; Dwiggins, S.; Katz, A.; Lin, X (1993). Information seeking in full-text end-user-oriented search-systems - The roles of domain and search expertise. Library and Information Science Research, 15 (1), 35-69.
95. Markey, K., Atherton, P (1978). Online training and practice manual for ERIC Database Searchers; ERIC Clearinghouse on Information Resources: Syracuse, NY
96. Marques, A. B., Rodrigues, R., & Conte, T. (2012, August). Systematic Literature Reviews in Distributed Software Development: A Tertiary Study. In Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on (pp. 134-143). IEEE.
97. Micarelli, A. Gasparetti, F; Sciarrone, F & Gauch, S (2007). Personalized search on the World Wide

Web. In Lecture Notes in Computer Science. Volume 4321 (the Adaptive Web), pp. 195-230, 2007.
98. Mislove, A; Gummadi, K. P & Druschel, P (2006). Exploiting social networks for internet search. In Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets'06).
99. Montenegro R. & Tetali P. (2006). Mathematical aspects of mixing times in Markov chains. Foundations and Trends in Theoretical Computer Science, 1(3), 237 – 354.
100. Moukdad, H.; Bulk, A (2001). Users' perceptions of the Web as revealed by transaction log analysis. Online Information Review, 25 (6), 349-359.
101. Novak, J. D. (1998). Learning, creating, and employing knowledge: Concept maps as facilitative tools in schools and corporations. Mahwah, NJ: Lawrence Erlbaum Associates.
102. Novak, J. D., & Gowin, D. B. (1984). Learning how to learn. New York, NY: Cambridge University Press.
103. Novak, J., Fleishmann, M., Strauss, W., Schneider, M., Wurst, M., Morik, K., & Kunz, C. (2002). Augmenting the knowledge bandwidth and connecting heterogeneous expert.
104. Okoli, C. and Schabram, K. (2010). A Guide to Conducting a Systematic Literature Review of Information Systems Research. Sprouts. Working Papers on Information Systems, 10(26).
105. P. A. Chirita, W. Nejdl, R. Paiu & C. Kohlschütter (2001). Employing odp metadata to personalize search. In Proceedings of the ACM Conference on Information Retrieval (SIGIR '01), pp. 250-257.
106. Palmquist, R. A., & Kim, K. S (2000). Cognitive style and online search experience on Web search performance. Journal of the American Society for Information Science and Technology, 51 (6), 558-567.
107. Pennanen, M., Vakkari, M, & Students', P. (2003). Conceptual structure, search process and outcome while preparing a research proposal. Journal of the American Society for Information Science, 54 (8), 759-770.
108. Prasad Chebolu & Pall Melsted (2008). "PageRank and the random surfer model", Proceedings of 19th annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1010-1018
109. Ready, R. C & Hu, D. (1995). Statistical Approaches to the Fat Tail Problem for Dichotomous Choice".
110. Rieh, S. Y.; Xie, H (2006). Analysis of multiple query reformulations on the web: The interactive information retrieval context. Information Processing & Management, 42 (3), 751-768.
111. Rosvall, M., & Bergstrom, C. T. (2010). Mapping change in bulk networks. PloS one, 5(1), e8694.
112. Rosvall, M., & Bergstrom, C. T. (2011). Multilevel compression of random walks on networks reveals hierarchical organization in bulk integrated systems. PloS one, 6(4), e18209.
113. S. Brin & L. Page (1998). The anatomy of a bulk-scale hypertextual Web search engine, In Proceedings of the International World Wide Web Conference (WWW '98), pp. 107-117, 1998. International Journal on Web Service Computing (IJWSC), 3(1),
114. Salton, G & Buckley, C (1988). Term-weighting approaches in automatic text retrieval. Information Processing and Management. 24(5), 513-523.
115. Saracevic, T (1996). Modeling interaction in information retrieval (IR): A review and proposal. Proceedings of the 59th ASIS Annual Meeting 1996, 33, 3-9.
116. Saracevic, T (1997). The stratified model of information retrieval interaction: Extension and applications. Proceedings of the 60th ASIS Annual Meeting 1997, 34, 313-327.
117. Schacter, J.; Chung, G. K. W. K.; Dorr, A (1998). Children's Internet searching on complex problems: Performance and process analyses. Journal of the American Society for Information

Science, 49 (9), 840-849.

118. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, X. & Weikum, G. (2008). Efficient top-k querying over social-tagging networks. In Proceedings of the ACM Conference on Information Retrieval (SIGIR '08), pp. 523-530, 2008.
119. Shiri, A. A.; Revie, C (2003). The effects of topic complexity and familiarity on cognitive and physical moves in a thesaurus-enhanced search environment. Journal of Information Science, 29 (6), 517-526.
120. Shute, S. J.; Smith, P. J (1993). Knowledge-based search tactics. Information Processing and Management, 29 (1), 29-45.
121. Siegfried, S.; Bates, M.J.; Wilde, D.M (1993). A profile of end-user searching behavior by humanities scholars: The Getty online searching project (Rep. No. 2). Journal of the American Society for Information Science, 44 (5), 273-291.
122. Silverstein, C.; Henzinger, M.; Marais, H.; Morica, M (1999). Analysis of a very bulk Web search engine query log. SIGIR Forum, 33 (1), 6-12.
123. Soloman, P (1993). Children's information retrieval behavior: A case analysis of an OPAC. Journal of the American Society for Information Science, 44 (5), 245-264.
124. Spink, A.; Jansen, B. J (2004). Web Search: Public Searching of the Web; Kluwer Academic Publishers: Boston.
125. Spink, A.; Wolfram, D.; Jansen, B. J.; Saracevic, T (2001). Searching the Web: The public and their queries. Journal of the American Society for Information Science, 52 (3), 226-234.
126. Suri, P. & Taneja, H. (March 2012). An Integrated Ranking Algorithm for Efficient Information Computing in Social Networks. International Journal on Web Service Computing (IJWSC), 3(1), 31 – 44.
127. Sutcliffe, A. G.; Ennis, M.; Watkinson, S. J (2000). Empirical studies of end-user information searching. Journal of the American Society for Information Science, 51 (13), 1211-1231.
128. Vakkari, P (2000). ECognition and changes of search terms and tactics during task performance: A longitudinal study. In RIAO' 2000 Conference Proceedings, Content-Based Multimedia Information, Collège de France, Paris, France, April 12-14, 2000; RIAO, Eds.; C.I.D.: Paris, 2000; 1, 894-907. http://www.info.uta.fi/vakkari/Vakkari_Tactics_RIAO2000.html (accessed July 15, 2018).
129. Vakkari, P (2000). Relevance and contributory information types of searched documents in task performance. In Proceedings of the 23rdAnnual International ACM SIGIR Conference on Research and Development in Information Retrieval; Belkin, N.J., Ingwersen, P., Leong, M-K, Eds; SIGIR forum; ACM Press: New York, Vol. 34, 2-9.
130. Vakkari, P. A (2001). Theory of the task-based information retrieval process. Journal of Documentation, 57 (1), 44-60.
131. Vakkari, P.; Hakala, N (2000). Changes in relevance criteria and problem stages in task performance. Journal of Documentation, 56, 540-562.
132. Vakkari, P.; Pennanen, M.; Serola, S (2003). Changes of search terms and tactics while writing a research proposal. Information Processing and Management, 39 (3), 445-463.
133. W. B. Frakes & R. Baeza-Yates. Information Retrieval: Data Structures & Algorithms, (2009).
134. Walker, G.; Janes, J (1999). Online Retrieval: A Dialogue of Theory and Practice, 2nd Ed.; Libraries Unlimited: Englewood, Colorado.
135. Wang, P.; Berry, M.; Yang, Y (2003). Mining longitudinal Web queries: Trends and

patterns. Journal of the American Society for Information Science and Technology, 54 (8), 743-758.

136. Wang, P.; Hawk, W. B.; Tenopir, C (2000). Users' interaction with World Wide Web resources: An exploratory study employing a holistic approach. Information Processing & Management, 36 (2), 229-251.
137. Webometrics Website (2016).
138. Wildemuth, B. M (2004). The effect of domain knowledge on search tactic formulation. Journal of the American Society for Information Science and Technology, 55 (3), 246-258.
139. Wilson, T. D (2000). Human information behaviour. Informing Science, 3(2), 49-56.
140. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Systematic Literature Reviews. Experimentation in Software Engineering, 45-54.
141. Wolfram, D.; Xie, H (2002). Traditional IR for Web users: A context for general audience digital libraries. Information Processing & Management, 38 (5), 627-648.
142. X. Shen., B.Tan, & C. Zhai, C (2002). Ucair: Capturing and exploiting context for personalized search. In Proceedings of the Information Retrieval in Context Workshop, SIGIR IRiX'05, 2005.
143. Xie, H (2000). Shifts of interactive intentions and information-seeking strategies in interactive information retrieval. Journal of the American Society for Information Science, 51 (9), 841-857.
144. Xie, H (2002). Patterns between interactive intentions and information-seeking strategies. Information Processing & Management, 38 (1), 55-77.
145. Xie, H. Understanding human-work domain interaction: Implications for the design of a corporate digital library. Journal of the American Society for Information Science and Technology, 57 (1), 128-143.
146. Xie, I. Interactive information retrieval in digital environments; IGI Global Inc.: Hershey, Pennsylvania, 2008.
147. Xing, W & Ghorbani, A. (2004). "Weighted PageRank algorithm". Proceedings of the Second Annual Conference on Communication Networks and Services Research 21-24 May 2004. Fredericton, Canada.
148. Y. Hu, G. Xin, R. Song, G. Hu, S. Shi, Y. Cao, & H. Li, H. (2005). Title extraction from bodies of html documents and its application to web page retrieval. In Proceedings of the ACM Conference on Information Retrieval (SIGIR '05), pp. 250-257.
149. Zareh Bidoki A. and Yazdani N. (2007). "DistanceRank: An intelligent ranking algorithm for web pages". Information Processing and Management (2007), doi:10.1016/j.ipm.2007.06.004