A Survey: Static and Dynamic Ranking

Aditi Sharma
Amity University
Noida, U.P.
India

Nishtha Adhao
Amity University
Noida, U.P.
India

Anju Mishra
Amity University
Noida, U.P.
India

ABSTRACT
The search engines are an important source of information. They work on mechanism of information retrieval. But the task does not end here. The bulk information retrieved has to be provided to the user as a list such that the best suited information lies at the top and so on. This process is called ranking. This paper is a review on different ranking algorithms broadly classified into static and dynamic ranking techniques.

General Terms
Information Retrieval, Web Search, Ranking Algorithms.

Keywords
Static Ranking, Dynamic Ranking, Information Retrieval, Ranking.

1. INTRODUCTION
Majority of internet users rely on search engines for extracting information by providing a query from any walk of life. These queries are processed by the search engines and a certain information retrieval or mining algorithm is applied to obtain the cluster of documents related to the query. After the retrieval of these documents, an important task is to present these documents in a list where documents at the top are the ones considered more relevant for the user. This task is called ranking of documents.

Ranking has been a topic of consideration by researchers for a long time and it still is. Earlier ranking algorithms were based on prior information about the websites. PageRank, HITS, SALSA, RankNet and fRank are examples of such algorithms. These use static features of web pages and thus are termed here as Static Ranking algorithms.

But the user is unaware of the structure of web. The queries are not concrete and specific to the structure of web pages. Thus, static algorithms fail sometimes, when they have to rank for ambiguous queries. To satisfy the user in better manner, the Dynamic Ranking algorithms came into picture. They provide results by taking into consideration the attributes of the query and interaction of user with the system. Fish Search, Diversified Ranking, Two-Level Ranking, Rank Refinement and Real-Time Implicit Feedback are Dynamic Ranking Retrieval Techniques.

In this paper, the survey and comparisons of algorithms under static and dynamic techniques are carried out.

2. STATIC RANKING
A search engine provides result of any user query. The users not only want the documents satisfying their query, but they also want the most relevant document to be placed at the top and least relevant at the bottom of the list. The earliest ranking model was Boolean model but it was not appropriate for end-user. Later, the vector and probabilistic models came. The vector model considered query and document as two vectors and calculated a cosine function to compute the similarity between the vectors. Higher the value of cosine function, higher the rank of document. Probabilistic model says that weight of query terms that appear in previously retrieved documents is higher than other documents as there is a high probability that previously retrieved document is more relevant to user query.

Static algorithms use these models to explore the document structure and rank them. We would now discuss few algorithms under this category.

2.1 PageRank
As said earlier, static algorithms are based on the structure of documents. PageRank is a ranking mechanism that was used by Google for a long time. This algorithm ranks different websites based on the number of their backlinks. Backlinks are the links that are pointed to a site from other sites. In the following figure, A, B, C, D, E are websites linked to each other.

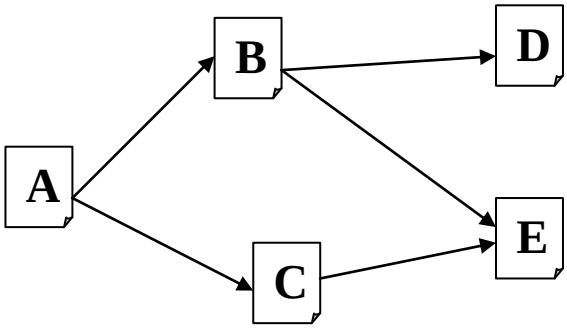

Fig 1: Backlinks

Here, the website A is backlink for B and C. Both B and C are backlinks for E. B is a backlink for D.

The PageRank algorithm states that PageRank of a website depends upon the PageRanks of all its backlinks. Thus, the page which has the highest sum of PageRank of its backlinks has the highest rank. A simple formula [1] for calculating PageRank of web pages E in the previous figure is –

$PR(E) = \sum_{v \in W} (PR(v)/L(v))$

PR(E) = page rank of E
W = set of all web pages linking to E (W={B,C})
PR(v) = page rank of all the elements in set W
L(v) = number of pages from page v (L(B) = 2 and L(C) = 1, since B links to D and E, C links to E)

2.2 Weighted PageRank
This algorithm was proposed by Wenpu Xing and Ali Ghorbani which is an extension of PageRank algorithm [5].

The pages are assigned rank values according to their importance, calculated in terms of weights to the incoming as well as outgoing links. assigns larger rank values to more important (popular) pages instead of dividing the rank value of a page evenly among its outlink pages.

Weight W(m,n)[in] is the weight of incoming links calculated on the basis of number of incoming links to page n and the number of incoming links to all reference pages of page m. Weight W(m,n)[out] is the weight of outgoing links calculated as number of outgoing links of page n and the number of outgoing links of all the reference pages of page m.

Then the weighted PageRank is given by formula –

$$WPR(n) = (1-d) + d \sum WPR(m)\, W(m,n)[in]\, W(m,n)[out]$$

2.3 Hyper-Link Induced Topic Search: HITS

This algorithm is based on following terms:

In-links: for a web page A, in-links are the hyperlinks pointing to A.

Out-links: for a page A, out-links are the hyperlinks present on A, pointing to some other webpage.

Authority: an authority is a web page with in-links.

Hubs: a hub is a web page with many out-links.

A page can be a hub or authority or both at the same time. It treats web as a directed graph with vertices as the pages and edges as the links between these pages. It has two steps:

1. Sampling Step:- In this step a set of relevant pages for the given query are collected.

2. Iterative Step:- In this step Hubs and Authorities are found using the output of sampling step.

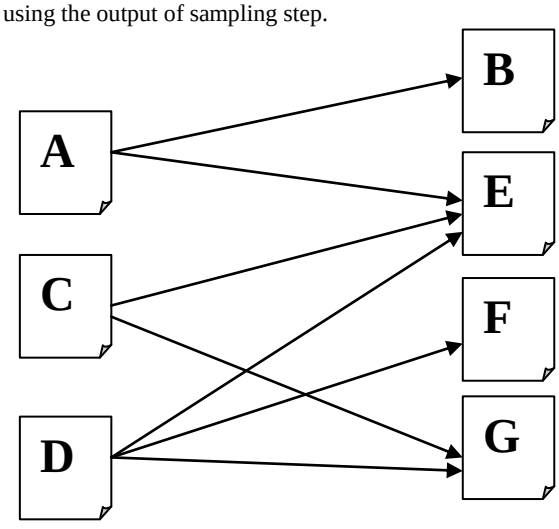

Fig 2: Hubs and Authorities

HITS forms a bipartite graph of all the retrieved pages. One side of the bipartite graph contains Hubs as its nodes, while the other side contains authority as its nodes. The authority weight of a page is proportional to the sum of hub weights of pages that link to it. Similarly a hub of a page is proportional to the sum of authority weights of pages that it links to.

This algorithm tries to restrict the retrieved documents to a size as small as possible with only most strong authorities, so that user is provided with only relevant results.

At first the web pages are selected by analysing their contents against the query, then the selected pages are analysed for their structure only [3].

The Hits algorithm can be stated as –

1.Let z denote the vector (1, 1, 1, ..., 1) ∈ R_n

2.Set $a_0 = z$

3.Set $h_0 = z$

4.For i = 1, 2, ..., k

4.1.Apply the I operation to (a_{i-1}, h_{i-1}), obtaining new a-weights a'_i

4.2.Apply the O operation to (a'_i, h_{i-1}), obtaining new h-weights h'_i

4.3.Normalize a'_i, obtaining a_i

4.4.Normalize h'_i, obtaining h_i

5.Return (a_k,h_k)

Algorithm 1: HITS

G: a collection of n linked pages
k: a natural number

2.4 SALSA

SALSA: The Stochastic Approach for Link-Structure Analysis, is a hybrid of PageRank and HITS algorithm. PageRank is a query independent link based ranking technique while Salsa is a query dependent link-based technique, like HITS. The approach is based upon the theory of Markov chains, and relies on the stochastic properties of random walks. It performs a random walk on hubs and authorities of bipartite graph [2]. If algorithm is at a hub node currently, then it chooses any out-link randomly and moves to corresponding authority node. Similarly, if algorithm is at authority node currently, it will choose an in-link randomly and move to a hub node.

Similar to HITS technique, Salsa also generates a neighbourhood graph with hubs and authorities. For each vertex, a hub and an authority score is calculated as their principal eigenvector. The in and out degree of each vertex is considered. The strategy states that, for the collection of pages P, retrieved for a topic T, the authoritative pages should be linked by many pages in the sub-graph built by P. Now, a random walk on this sub-graph will visit authority pages with high probability.

If an authority graph contains more than one component, then algorithm selects a node at random and performs the random walk in the component containing that node. In HITS, hubs broadcast their weights to the authorities and authorities sum up the selective weights of the hubs that point to that authority. SALSA makes a variation to this operation. Instead of broadcasting, each hub divides its weight equally among the authorities to which it points and each authority divides its weight equally among the hubs that points to it [2]. Thus the relative authority of a node within a connected component of

graph is determined from local links, not from the structure of the component.

3. DYNAMIC RANKING

As discussed earlier, the static ranking don't take into consideration the interaction with user and faces issues like query ambiguity and diversity in intent of user. There is an inherent trade-off between number of results provided for user intent and number of intents retrieved [6]. Dynamic Ranking provides a way to combine the otherwise contradictory goals of result diversification and high recall [8].

These algorithms interact with the user to know his intent amongst the various possible intents, or they try to reorder the results of first retrieval process and provide refined results to the user. They focus on both the relevance and diversity. [6][9][11][12] are the ways in which dynamic ranked retrieval is obtained. Now we will discuss these algorithms in detail.

3.1 Two-Level Ranking using Mutual Information

Classification of a word depends not only on its meaning but also its relation with other words. This relation between words is called mutual information. Mutual information helps to understand the relevance of results to the user query [7].

Mutual information is calculated by exploring the relationship between joint probability of two words with their individual probability. It follows the thought that if two words are related to each other their mutual probability will be much higher than the probability of each of the words. An example of related words is book, pages, cover, author, and publisher.

This model works in two phases, called levels – retrieval and reordering. Two modules are used for this model - terms indexing module and, mutual and document terms information construction module.

3.1.1 Level-1: *Retrieval of Documents*

The first step is to retrieve the relevant documents. It makes use of the existing static retrieval method called Vector based document retrieval using Vector Space Model. It accepts a user query and retrieves the documents calculated to be similar to the query [14]. The documents are ranked by descending order of the weights of the retrieved documents. This information is contained in the indexing module.

At this level, the mutual information is also calculated using Mutual information construction module. A simple algorithm for calculation mutual information, as given by HYUN-KYU KANG [7], is –

1. For all documents

    1.1. Extract entry (title) of a document

    I .2. Extract <see> word or <see-also> words

        1.2.1. If there is <see> word, insert a pair of the entry and the <see> word and a pair of the <see> word and the entry

        1.2.2. If there are <see-also> words, insert a pair of the entry and the <see-also> word and a pair of the <see-also> word and the entry

    1.3. for all extracted terms

        1.3.1. Insert a pair of the entry and extracted the term and a pair of the extracted term and the entry

        1.3.2. If there are <see-also> words, insert a pair of the extracted term and the <see-also> word and a pair of the <see-also> word and the extracted term

        1.3.3. Insert a pair of the extracted term and the extracted term that is within 5 (window size) in a sentence and the reverse pair

2. Sort the pair of terms

3. Calculate the term frequency, the frequency of co-occurrence pair terms, and the total terms frequency

4. Calculate the mutual information value of co-occurrence pair terms

5. Sort the terms and descending by order of the mutual information value, and construct the index file and the posting file for the mutual information

Algorithm 2: Mutual Information Construction

The document information module contains the following information-

- Document identification
- Number of document terms
- Pair of terms
- Frequency of word and pairs

3.1.2 2) Level-2: Reordering of retrieved documents

This step is used to provide an optimized query result. A re-ordering engine is used to rank the documents retrieved at first level. It takes as input the mutual information and document terms information calculated in the first level.

Reordering is done via different formulas for calculating the similarity value of every document with respect to the query.

3.2 Dynamic Ranking Tree

Static ranking techniques use a probability model for dealing with ambiguous queries. Though this model works well for unambiguous query, their ranked results do not perform well for ambiguous query as they do not take into consideration the intentions of user [15]. To keep track of what user actually wants to see, rank tree is formed dynamically. Every node of this tree is a retrieved document and every path of the tree is the expected sequence of documents that a user would traverse to satisfy their intent.

An example of a query where the user might want to explore different topics related to it is "brown". When this query is given to a search engine, the different documents retrieved are – Brown University, meaning of brown as a colour, Browns fashion magazine, website of Dan Brown, Brown's hotels, Jerry Brown the Governor. All these retrieved documents

have no connection with each other and the user might follow just one of these documents. All these retrieved documents are nodes at the first level and documents related to these nodes form the child nodes on the corresponding paths.

The dynamic ranking tree allows the user to choose any of these nodes and traverse down the path starting from that node. Thus the user will be provided with more relevant documents after recording the first interaction.

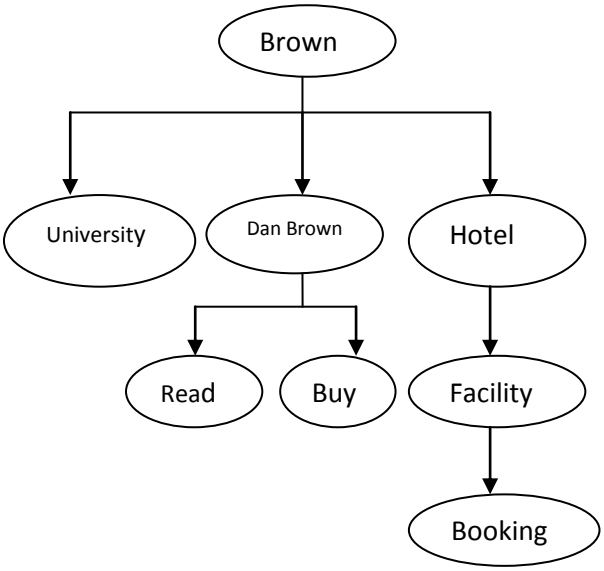

Fig 3: Rank Tree

It states two algorithms: DynamicMyopic and DynamicLookahead. DynamicMyopic algorithm takes greedy approach. After every iteration it adds a document with highest gain in utility, given by d. It is similar to sorting of documents with respect to their utility gain.

The DynamicLookahead algorithm makes use of a lookahead estimation. The Myopic algorithm makes use of only the utility gain of the document, but Lookahead algorithm uses two approximate utility values of the two subtrees of the document into consideration. Thus, a document is chosen such that it not only results into maximum utility gain but also maximizes the utility of its two subtrees.

3.3 Two Level Dynamic Ranking

As we have depicted earlier in figure 1, a query provided by the user can have multiple intents. Conventional ranking algorithms rank the results by maximizing the probability of relevance independently for each document [16] and thus they prefer documents with most prevalent intention. In order to provide the best search result, it is better to first understand the intent of the user. For this, algorithms have been devised, called diversification based algorithms. Such algorithms include at least one result for as intents as possible as given in [17].

In order to remove this limitation, the authors of [6] have provided a method through which ranking would be performed at two levels –

1. First level results provide a list of diverse ranked documents.

2. Second level provides results related to the intent, as shown by the user's interaction with the first level results.

In this method the second level results depend upon the first level heads but it still provides the flexibility to users to track back to another document at the first level. The dynamic ranking methods given by [7] and [8] lack this flexibility.

This technique provides better result as it doesn't depends on the users to provide feedback at every level. But still it is a type of interactive retrieval, where user provides a feedback through the results provided at the first place, and this feedback is used by the system to again retrieve and rank the documents. The ranking uses a User Model which assumes that user will provide feedback at only the first level and that the user can return to first ranked results.

This is also a greedy algorithm. It calculates performance measure in the form of utility Ug(Θ|t), where g is a concave, positive, non-decreasing function, Θ is used for dynamic ranking and t is the user intent.

$$U_g(\Theta|t) = g\left(\sum_{i=1}^{|\theta|}\left(\gamma_i U(d_{i0}|t) + \sum_{j=1}^{|\theta i|}\gamma_{i,j} U(d_{i0}|t)\, U(d_{ij}|t)\right)\right) \quad (1)$$

Where, di are documents and γ are position dependent discount factors which decrease with position in ranking [6].

For a second level document, the utility value is set to zero. If and only if the head document in first level has been assigned non-zero relevance, then the corresponding document at second level will have non-zero utility for the intent related to the head document.

The diversity of intents depends on the function g. The steeper the function is more will be the diversity of the documents ranked. For a query q and set of documents D, the possible intents for the query T(q) and their distribution P(t|q), the algorithms forms a ranking matrix such that the utility is maximized. The matrix formed will be of the size LxW, where L is for length and W is for width. For every candidate row, the algorithm adds W documents which would result in maximum utility for that row. This continues till L rows have been processed.

3.4 GenDer

This is a generic diversified ranking algorithm [9]. It tells how to provide results catering to the different possible needs of the user. The algorithm described under this technique diversifies the top k-ranked documents.

The paper [9] introduced arbitrary relevance function and arbitrary similarity function. Using these two parameters the ranking is done. Diversity is a key factor to address the uncertainty and ambiguity in an information retrieval system. It is also an effective way to cover different aspects of information requirement [18]. Many diversification based algorithms have been centred on the extent of topic coverage in the result, or diversification of resultset.

The ranking algorithms measure their performance based upon the relevance or similarity matrices that are dependent on topics related to query and documents. The GenDer algorithm considers the relevance to the query as well as the diversification of the results as the main factors. There is always a trade-off between relevance and diversity. If an algorithm focuses on relevance, it can miss out some documents with lesser prominence but possible relevance to intent of user. On the other hand, when diversity is targeted, the numbers of relevant documents actually required by the user are missed out as the system focuses upon providing as many topics as possible. To take care of this trade-off,

GenDer uses a regularization parameter (w) is used to maintain balance between relevance and diversity. It also specifies that it is not possible to find the perfect balance, thus it provides a near optimal solution.

Notations used [9]:

X: set of n candidate documents

S: similarity matrix of size n x n. It is symmetric matrix

r(): ranking function. It returns relevance value for each document in X

T: subset of X. It has k elements. The goal of this technique is to find this subset T.

q: nx1 reference vector. Calculated as q=S.r. ith element of q gives the importance for rank of ith element in X.

w: regularization parameter that defines trade-off between relevance to query and diversification among the set of documents.

g(T): goodness function to calculate how good a document is in terms of both relevance and diversity.

argmax|T| = k g(T) = w$\sum_{i\in T} q$iri – $\sum_{i,j\in T} r$iSi,jrj

4. COMPARISON

Every algorithm comes with some advantages as well as some limitations. The performances of different algorithms can be judged on the basis of some measures like – concept used, merits and limitations. Here is a small table depicting the comparison between static and dynamic style of ranking.

Table 1: Static Ranking versus Dynamic Ranking

STATIC RANKING	DYNAMIC RANKING
A. Criteria : Basic Concept	
• Mining based on link structure. • Pages are ranked based on relative position of links or tags, number of links to or from the page, or relative distance between pages.	• Mining based on document's relevance to the query and rank using the information of user's interaction with the system.
B. Criteria : Relation to Query	
• Query-independent. • Remains static after query has been sent. • Results depend upon the link structure of the relevant pages.	• Query-dependent. • Every phase depends on the query and the information need of the user.
C. Criteria : Merits	
• Simple, fast and easy to perform. Based on relevancy of the document. • Provide very good results especially in the case of unambig-uous query.	• More accurate with respect to fulfilling the needs of the user. Does not depend on some pre-set indexing, but works dynamically for every query. • Developed with unamb-iguous nature of query in mind.
D. Criteria : Limitations	
• Not alive after receiving the query. It checks the probability of the relevance of document with the query and produces results. • Lacks maintaining diversity of result along with the relevance.	• Remains active even after producing results to the user. Keeps track of the user's selection amongst the produced documents and provide the documents relevant to the interest shown by the user. • Some algorithms are able to handle the trade-off between relevance and diversity, up to some extent

5. CONCLUSION

In this survey paper we have shown different techniques under the static and dynamic ranking mechanisms. It provides an overview of how different concepts are used and optimized for providing relevant results to the user for any information need. Through this study we have come across different aspects, advantages over other algorithms and limitations. Though lot of research has been done in the field of ranking, there is still immense scope available, since none of the ranking provides 100% percent relevance to the user information requirement for different queries.

6. REFERENCES

[1] S.Brin, L.Page, "The anatomy of a large-scale hyper textual web search engine", Proceedings of the 7th International World Wide web Conference, 1998

[2] Alessio Signorini, "A Survey of Ranking Algorithms", Department of Computer Science, University of Iowa, 2005.

[3] Jon Kleinberg, "Authoritative Sources in a Hyperlinked Environment", In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 1998.

[4] Dilip Kumar Sharma and A.K. Sharma, "A Comparative Analysis of Web Page Ranking Algorithms", International Journal on Computer Science and Engineering, Vol. 02, No. 08, 2010, 2670-2676.

[5] S. Madria, S. S. Bhowmick, W. K. Ng, and E.P. Lim, "Research issues in web data mining". In Proceedings of the Conference on Data Warehousing and Knowledge Discovery, pages 303–319, 1999.

[6] Karthik Raman, Thorsten Joachims and Pannaga Shivaswamy, "Structured learning of two-level dynamic rankings", Proceedings of the 20th ACM international conference on Information and knowledge management, pages- 291-296, 2011

[7] Hyun-Kyu Kang and Key-Sun Choi, "Two-Level Document Ranking Using Mutual Information In Natural Language Information Retrieval", Information Processing & Management, Vol. 33, No. 3. pp. 289-306. 1997

[8] Christina Brandt, Thorsten Joachims, Yisong Yue and Jacob Bank, "Dynamic Ranked Retrieval", Proceedings of the fourth ACM international conference on Web search and data mining, Pages 247-256, 2011

[9] Jingrui He, Hanghang Tong, Qiaozhu Mei and Boleslaw K. Szymanski, "GenDer: A Generic Diversified Ranking Algorithm", Advances in Neural Information Processing System 25, edited by P.Barlett, F. Pereira, L. Bottou, C. Burges and K. Weinberger, NIPS, page 1151-1159, 2012

[10] Neelam Duhan, A. K. Sharma, Komal Kumar Bhatia, "Page Ranking Algorithms: A Survey", Advance Computing Conference, IACC 2009 IEEE International, 2009

[11] Chieh-Jen Wang, Yung-Wei Lin, Ming-Feng Tsai and Hsin-His Chen, "NTU Approaches to Subtopic Mining and Document Ranking at NTCIR-9 Intent Task", Proceedings of NTCIR Workshop Meeting, 2011

[12] Rong Jin, Hamed Valizadegan and Hang Li, "Ranking Refinement and Its Application to Information Retrieval", WWW 2008 / Refereed Track: Search - Ranking & Retrieval Enhancement, Beijing, China, April 21-25, 2008

[13] G. Salton, and C. Buckley, "Improving retrieval Performance by relevance feedback", Journal of the American Society for Information Science, 4/(4), 288-297, 1990

[14] G. Salton, & M. J. McGill, "Introduction to modern information retrieval", New York: McGraw-Hill, 1983

[15] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results", In ACM Conference on Web Search and Data Mining (WSDM), 2009.

[16] S. Robertson, "The probability ranking principle in information retrieval", Journal of Documentation, 33(4):294-304, 1977.

[17] Y. Yue and T. Joachims, "Predicting diverse subsets using structural svms", In International Conference on Machine Learning (ICML), 2008.

[18] F. Radlinski, P. N. Bennett, B. Carterette, and T. Joachims, "Redundancy, diversity and interdependent document relevance", SIGIR Forum, 43(2):46–52, 2009.