International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

Improving Information Retrieval Performance

Abhay Dwivedi[1], Ankit Maurya[2], Sandhya Rawat[3]
[1]Department of BCA, Shri L.B.S. Degree College, Gonda, Uttar Pradesh, India
[2]Department of Mathematics, Shri L.B.S. Degree College, Gonda, Uttar Pradesh, India
[3]I.E.T Dr. R.M.L.A. University Ayodhya, Uttar Pradesh, India

Article Info
Publication Issue :

Volume 8, Issue 5
September-October-2022
Page Number : 52-63

Article History
Accepted: 01 Sep 2022
Published: 09 Sep 2022

ABSTRACT

Locating interesting information is one of the most important tasks in Information Retrieval (IR). An IR system accepts a query from a user and responds with a set of documents. Generally, the system returns both relevant and non-relevant material and a document organization approach are applied to assist the user in finding the relevant information in the retrieved set. The two most widely used document organization approaches are the ranked list and clustering of the retrieved documents. Both these techniques have their strengths and weaknesses.

This paper addresses the problem of offering scalable, adaptive, efficient, full-fledged information retrieval method. We consider the problem of combining ranking results from various sources. In the context of the Web, the main applications include building meta-search engines, combining ranking functions, selecting documents based on multiple criteria, and improving search precision through word associations. We develop a set of techniques for the rank aggregation problem and compare their performance to that of well-known methods. A primary goal of our work is to design rank aggregation techniques for providing robustness of search in the context of web.

Keywords:- Information Retrival (IR), meta search engine, rank aggregation

I. INTRODUCTION

This thesis addresses the problem of offering scalable, adaptive, efficient, full-fledged information retrieval method. We consider the problem of combining ranking results from various sources. In the context of the Web, the main applications include building meta-search engines, combining ranking functions, selecting documents based on multiple criteria, and improving search precision through word associations. We develop a set of techniques for the rank aggregation problem and compare their performance to that of well-known methods. A primary goal of our work is to design rank aggregation techniques for providing robustness of search in the context of web.

Locating interesting information is one of the most important tasks in Information Retrieval (IR). An IR system accepts a query from a user and responds with

a set of documents. Generally, the system returns both relevant and non-relevant material and a document organization approach are applied to assist the user in finding the relevant information in the retrieved set. The two most widely used document organization approaches are the ranked list and clustering of the retrieved documents. Both these techniques have their strengths and weaknesses. We begin by putting our work in the context of the previous research done in the field of Information Retrieval. We then show how automatically Combine Search and Ranking Results can be used in a novel and effective way. We define the evaluation methodology that we use to evaluate our approach. We then show how our approach can be explained to the user in a clear and intuitive fashion by presenting him or her with a clear visualization. We describe the rank aggression technique and hypothesize that it is indeed an intuitive way to navigate the retrieved set, after which we present the result of a small user study that supports our hypothesis. We conclude with the discussion of the results and an outline of directions for future work.

Specifically, we study the rank aggregation problem in the context of the Web, where it is complicated by a plethora of issues. We begin by underscoring the importance of rank aggregation for Web applications and clarifying the various characteristics of this problem in the context of the Web. We provide the theoretical underpinnings for stating criteria for "good" rank aggregation techniques and evaluating specific proposals, and we over novel algorithmic solutions. Our experiments provide initial evidence for the success of our methods, which we believe will significantly improve a variety of search applications on the Web.

II. DEFINITION

"Information retrieval deals with the representation, storage, organization of, and access to information items."

"IR is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."

An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines.

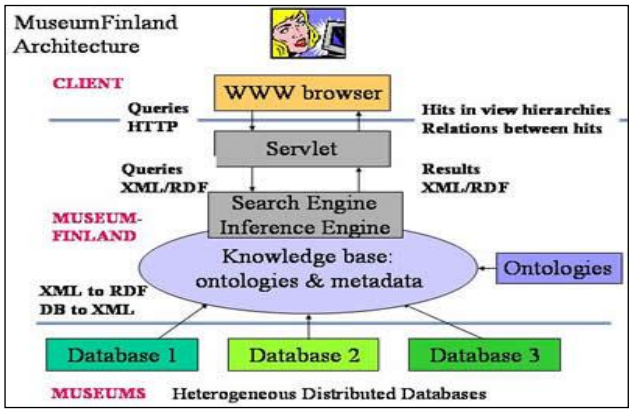

Figure 1: INFORMATION RETRIEVAL ON WEB

III. PROBLEM STATEMENT

3.1. Web IR problem and Open Research Issues:

- **The distribution of the web content**.

As a distinction between classic IR and web IR, the distribution of the web data all over the world makes it difficult to gather such data, and to overcome network limitations (e.g. bandwidth) and platform incompatibility.

- **The high data volatility**.

Every single day, millions of web pages are added while others are eliminated from the web. In addition, dangling links and domain name changes are also other web related issues which require more design effort.

- **The heterogeneity and size of the web data.**

The web content varies in terms of the languages in which web documents are written, the formats of the files being posted, and the media required to use the content. Moreover, the size of the whole web is in terabytes.

- **The lack of structure and data redundancy.**

The problem of the lack of structure occurs essentially because of the fact that the web most used Hyper textual language (HTML) does not impose restrictions on its document structure. A standard mechanism to deal with the text provided within HTML documents is far from being reached. In addition, the redundancy or repetition of the same content on different sites by mirroring or proxy servers is also another critical issue for which search engines have to dedicate extra effort. A study showed that approximately 30% of web pages are duplicated.

- **Poor content quality.**

This is due to the types in web documents in addition to the poor interpretation of languages other than English. In addition, the web is an open Web Information Retrieval and medium on which anybody can post whatever they want with no editorial processes.

- **Web traps**

As discussed in there are several web traps that a web crawler might encounter and have to deal with. Among which is the anti-spam protocols (page flooding), URL aliases, content duplication (page mirroring), and artificial infinite paths creators that get web crawlers into infinite loops of fetching. In addition, the limited workload possible of the DNS servers stands as another obstacle around which a web crawler has to work. The research interests in web information retrieval involve improving several search factors. , the most important issues accompanying web IR research can be summarized in the following discussion. However, since there have been several years since this source of information was presented in the public literature, the discussion will go through only those issues that have not been completely resolved yet.

- **Modeling the web.**

The current working models, such as the vector space, are fairly exhausted. Finding a better way to cope with the undeniable growth of the web content is a matter of concern to web IR researchers.

- **Querying**.

Embedding structure in search queries is a new idea that is being developed for better search accuracy. In addition, data mining techniques have potential improvements to deal with the different kinds of data containers being posted on the web.

- **Distributed architecture.**

There have been several attempts to replace the current indexing mechanisms with more effective search agent based techniques. Those are intended to traverse the web and have the data pushed to the search engine core process. Cooperation between metasearch engines is also utilized to increase scalability and to accommodate the web growing content.

- **Ranking**.

Most of the introduced ranking techniques rely on exploiting the content of web pages in addition to the hyperlinks of the web graph. However, there is a research tendency towards utilizing the structure of web documents and the distribution of their elements in the ranking process. This is meant to provide better relevancy in search results. Furthermore, some ideas suggest integrating the user in the search process, such as by using profiles on which ranking the resulted set of documents will be based.

- **Indexing**

Manning et al. [1] describe the basic process of creating an inverted index. The process takes a list of normalized tokens for each document as input. The most important steps in index construction is the sorting and grouping of the terms. In the simplest case, terms are sorted alphabetically and multiple occurrences of the same term in a document are merged. Instances of the same term across documents are then grouped together, and the resulting list of terms and their occurrences is split into dictionary and postings. In addition to a pointer to the posting list, each term in the dictionary can contain certain pieces of statistical information such as document frequency, used in ranked retrieval models. The postings are secondarily sorted by

document number to allow efficient query processing. The dictionary file is much smaller than the postings file and is usually kept in memory to optimize response time. Various data structures have been proposed to optimize storage and access efficiency of the dictionary and posting lists. In addition, looking for better indexing techniques to deal with the different types of documents, such multimedia files, is a matter of focus for more efficient web search.

- **The hidden web.**

The hidden web refers to web pages created on demand in a dynamic way that is not reachable by the current working crawlers. Migrating search agents have the potential to find and retrieve such documents since the creation of those pages happens in the presence of such agents.

- **Browsing and presenting.**

Visualization techniques provide further enhancements to the user's understanding of the search results. Therefore, several attempts have been done in both visualizing the data mining process and the rendering process. Improvements in both phases of the search procedure have been aimed at enhancing the speed of finding the intended answer set in the returned results.

- **Clustering**

Clustering algorithms group a set of documents into subsets or clusters. The algorithms' goal is to create clusters that are coherent internally, but clearly different from each other. In other words, documents within a cluster should be as similar as possible; and documents in one cluster should be as dissimilar as possible from documents in other clusters.

Therefore, any method for rank aggregation for Web applications must be capable of dealing with the fact that only the top few hundred entries of each ranking are available. Of course, if there is absolutely no overlap among these entries, there isn't much any algorithm can do; the challenge is to design rank aggregation algorithms that work when there is limited but non-trivial overlap among the top few hundreds or thousands of entries in each ranking. Finally, in light of the amount of data, it is implicit that any rank aggregation method has to be computationally efficient.

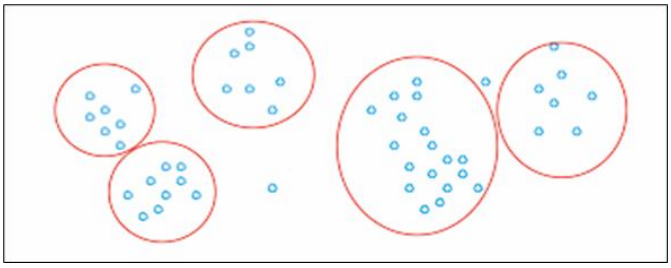

Figure 2 AN EXAMPLE OF A DATA SET WITH A CLEAR CLUSTER STRUCTURE

IV. LITERARTURE REVIEW

C.D. Manning, P. Raghavan, and H. Schutze,2008
Class-tested and coherent, this groundbreaking new textbook teaches web-era information retrieval, including web search and the related areas of text classification and text clustering from basic concepts. Written from a computer science perspective by three leading experts in the field, it gives an up-to-date treatment of all aspects of the design and implementation of systems for gathering, indexing, and searching documents; methods for evaluating systems; and an introduction to the use of machine learning methods on text collections.

IMPLEMENTING

Adi Wahyu Pribadi, Zaenal Arifin Hasibuan, 2003
Information Retrieval is concerned with selecting documents from a collection that will be of interest to a user with a stated information need or query. This paper describes a retrieval model that uses probabilistic inference networks consisting of a document network which is built once to represent document collections and a query network which is built every new query or information need is given. In the first section of this paper, an inference network model will be introduced and described briefly. Next, a simple example is given to illustrate how inference networks works on document collections. In the

following section, the model will be implemented using news articles taken from Republika Online and Tempo Interaktif. Boolean and extended Boolean models are given to compare the proposed model. In the end, the paper will conclude that inference networks will be able to develop technique that can improve performance over conventional retrieval models. The model also has limitations and remains open to examine and study further.

Liu, X. & Croft, W.B. 2004, Statistical language models have recently been successfully applied to many information retrieval problems. A great deal of recent work has shown that statistical language models not only lead to superior empirical performance, but also facilitate parameter tuning and open up possibilities for modeling nontraditional retrieval problems. In general, statistical language models provide a principled way of modeling various kinds of retrieval problems. The purpose of this survey is to systematically and critically review the existing work in applying statistical language models to information retrieval, summarize their contributions, and point out outstanding challenges.

Ponte, J. & Croft, W.B. 1998, This article surveys recent research in the area of language modeling (sometimes called statistical language modeling) approaches to information retrieval. Language modeling is a formal probabilistic retrieval framework with roots in speech recognition and natural language processing. The underlying assumption of language modeling is that human language generation is a random process; the goal is to model that process via a generative statistical model.

Han, J. and Kamber, M. 2001 The course explores the concepts and techniques of data mining, a promising and flourishing frontier in database systems. Data Mining is automated extraction of patterns representing knowledge implicitly stored in large databases, data warehouses, and other massive information repositories. It is a decision support tool that addresses unique decision support problems that cannot be solved by other data analysis tools such as Online Analytical Processing (OLAP). The course covers data mining tasks like constructing decision trees, finding association rules, classification, and clustering. The course is designed to provide students with a broad understanding in the design and use of data mining algorithms. The course also aims at providing a holistic view of data mining. It will have database, statistical, algorithmic and application perspectives of data mining.

Meng, W., Yu, C., & Liu, K.-L.,With the increase of the number of search engines and digital libraries on the World Wide Web, providing easy, efficient, and effective access to text information from multiple sources has increasingly become necessary. In this article, we presented an overview of existing meta search techniques. Our overview concentrated on the problems of database selection, document selection, and result merging. A wide variety of techniques for each of these problems was surveyed and analyzed. We also discussed the causes that make these problems very challenging. The causes include various heterogeneities among different component search engines due to
the independent implementations of these search engines, and the lack of information about these implementations because they are mostly proprietary.

Aslam, J. A., Montague, M. 2001 Given the ranked lists of documents returned by multiple search engines in response to a given query, the problem of metasearch is to combine these lists in a way which optimizes the performance of the combination. This paper makes three contributions to the problem of metasearch: (1) We describe and investigate a metasearch model based on an optimal democratic voting procedure, the Borda Count; (2) we describe and investigate a metasearch model based on Bayesian inference; and (3) we describe and investigate a model for obtaining upper bounds on the performance of metasearch algorithms. Our experimental results show that metasearch algorithms based on the Borda and Bayesian models usually outperform the best

input system and are competitive with, and often outperform, existing metasearch strategies.

Cynthia Dwork 2001 The rank aggregation problem is to combine many different rank orderings on the same set of candidates, or alternatives, in order to obtain a "better" ordering. Rank aggregation has been studied extensively in the context of social choice theory, where several "voting paradoxes" have been discover.

Salton, G., Fox, E. & Wu, H. 1983 The Extended Boolean models have give much better performance than the standard Boolean model The standard boolean retrieval system does not provide ranked retrieval output because it cannot compute similarity co-efficients between queries and documents. Extended boolean models like fuzzy set, Wailer-Kraft, P-Norm and Infinite-One have been proposed in the past to support ranking facility for the boolean retrieval system. The behavior of the previous extended boolean models and we address important mathematical properties to affect retrieval effectiveness. As we know that retrieval process can be effective by this process it improves method of retrieval and make it effective by adding important facts or weights that make it better than standard boolean method.

Sparck Jones, K. & Willett, P. (Eds). 1997 Reading in information retrieval is big issue to resolve problem of read and fetch information because past method of retrieval from library system was tough and time taken. Readings in Information Retrieval attempts to help the library world catch up with the present and prepare for a future in which automation plays an important role, but the past achievements of librarianship, among them the development of accurate and understandable bibliographic data, should not be trampled underfoot in the surge forward; they are perhaps more important now than ever before. We have to make effective and improved system for retrieve information that we can read it easy and inappropriate way.

Strzalkowski, T., Tzukermann, E. & Klavans, J. 2002 Information Retrieval remains one of the most challenging problems in NLP. In information retrieval Many Natural Language Processing (NLP) techniques have been used . The results are not encouraging. Simple methods like; stop wording, porter-style stemming, etc. usually yield significant improvements, while higher-level processing like chunking, parsing, word sense disambiguation, etc. only yield very small improvements or even a decrease in accuracy. At the same time, higher-level methods increase the processing and storage cost dramatically. This makes them hard to use on large collections. We review NLP techniques and come to the conclusion that NLP needs to be optimized for IR in order to be effective.

Estivill-Castro, V. and Yang, J. 2000 General purpose and highly applicable clustering methods are required for information retrival. k-Means has been adopted as the prototype of iterative model-based clustering because of its speed, simplicity and capability to work within the format of very large databases. However, k-MEANS has several disadvantages derived from its statistical simplicity. We propose algorithms that remain very efficient, generally applicable, multidimensional but are more robust to noise and outliers. We achieve this by using medians rather than means as estimators of centers of clusters. Comparison with k-Means, EM and Gibbs sampling demonstrates the advantages of our algorithms.

Dhillon I. and Modha D. 2001 Retrieval from large collection of data is effective when we used clustering k method for information retrival. Unlabeled document collections are becoming increasingly common and available; mining such data sets represents a major contemporary challenge. Using words as features, text documents are often represented as high-dimensional and sparse vectors–a few thousand dimensions and a sparsity of 95 to 99% is typical. The algorithm outputs k disjoint clusters each with a concept vector that is the centroid of the cluster normalized to have unit Euclidean norm. As our first contribution, we empirically demonstrate

that, owing to the high-dimensionality and sparsely of the text data, the clusters produced by the algorithm have a certain "fractal-like" and "self-similar" behavior.

Fisher, D., 1987 Conceptual clustering is an important way of summarizing and explaining data. However, the recent formulation of this paradigm has allowed little exploration of conceptual clustering as a means of improving performance. Furthermore, previous work in conceptual clustering has not explicitly dealt with constraints imposed by real world environments. This article presents COBWEB, a conceptual clustering system that organizes data so as to maximize inference ability. Additionally, COBWEB is incremental and computationally economical, and thus can be flexibly applied in a variety of domains.

V. AN APPROCH TO IMPROVE THE PERFORMANCE OF INFORMATION RETRIEVAL

In every query formulation technique there is a human in the loop. From very simple queries to extremely complex queries and there must be a person to define the information need in the form of a query. One of the system performance measures that are often ignored is the level of effort required for query construction. In many cases of the information need, the required query is quite simple. Specifically, simple queries perform well in the case where the information density is high. For example, if the analyst wants to know the score of the Lakers game last night, there are many sources that can provide that information and a simple query will suffice. In other cases, particularly where the information density low, the query must be complex and broad so that relevant data is not missed.

An online information seeker often fails to find what is wanted because the words used in the request are different from the words used in the relevant material. Moreover, the searcher usually spends a significant amount of time reading retrieved material in order to determine whether it contains the information sought. The conceptual indexing and retrieval system used for these experiments automatically extracts words and phrases from unrestricted text and organizes them into a semantic network that integrates syntactic, semantic, and morphological relationships. The resulting conceptual taxonomy is used by a specific passage-retrieval algorithm to deal with many paraphrase relationships and to find specific passages of text where the information sought is likely to occur. The database systems support a simple Boolean query retrieval model, where a selection query on a SQL database returns all tuples that satisfy the conditions in the query. This often leads to the Many-Answers Problem: when the query is not very selective, too many tuples may be in the answer [36].

Document surrogates containing both anchor text and query associations have been found to improve retrieval effectiveness. Indeed, Web search engines have long made use of anchor text to improve result quality. For retrieval purposes, a text document may be supplemented with additional terms derived from external sources such as metadata, anchor text and so on. In the case of document surrogates, the additional terms form their own document which is used instead of the original. Retrieval may be based on scoring the surrogate collection or those scores may be combined with scores from the original collection. The following are examples of the use of surrogate or supplemented documents [36].

Given a universe U, an ordered list (or simply, a list) L with respect to U is an ordering of a subset S of U, i.e. -

L = [x1 > x2 > ... > xd], with each xi in S, and > is some ordering relation on S. Also, if i in U is present in L, let L(i) denote the position or rank of i (a highly ranked or preferred element has a low-numbered position in the list). For a list L, let |L| denote the number of elements. By assigning a unique identifier to each element in U, we may assume without loss of generality that U = {1, 2,..., |U|}.

Depending on the kind of information present in L, three situations arise -

1. If L contains all the elements in U, then it is said to be a full list. Full lists are, in fact, total orderings of U. For instance, if U is the set of all pages indexed by a search engine, it is easy to see that a full list emerges when we rank pages with respect to a query according to a fixed algorithm [41].

2. There are situations where full lists are not convenient or even possible. For instance, let U denote the set of all Web pages in the world. Let L denote the results of a search engine in response to some fixed query. Even though the query might induce a total ordering of the pages indexed by the search engine, since the index set of the search engine is almost surely only a subset of U, we have a strict inequality |L| < |U|. In other words, there are pages in the world which are unranked by this search engine with respect to the query. Such lists that rank only some of the elements in U are called partial lists.

A special case of partial list is as follows –

If S is the set of all the pages indexed by a particular search engine and if L corresponds to the top 100 results of the search engine with respect to a query, clearly the pages that are not present in list L can be assumed to be ranked below 100 by the search engine. Such lists that rank only a subset of S and where it is implicit that each ranked element is above all unranked elements, are called top d lists, where d is the size of the list [43].

To measure the distance between two full lists with respect to a set S, distance measures are -

(1) The distance D_1 is the sum, over all elements i in S, of the absolute difference between the rank of i according to the two lists. Formally, given two full lists L and M, their distance D_1 is given by-

D_1 (L, M) = $\sum_i$ $|L_i - M_i|$ (1)

After dividing this number by the maximum value (1/2) $|S|^2$, one can obtain a normalized value of the distance (D1), which is always between 0 and 1. The distance (D1) between two lists can be computed in linear time.

(2) The second distance (D2) counts the number of pair wise disagreements between two lists; that is, the distance between two full lists L and M is -

D_2 (L, M) = $|\{(i, j) : i < j, L_i < L_j$ but $M_i > M_j|$ (2)

Dividing this number by the maximum possible value (1/2) S (S - 1) we obtain a normalized version of the distance D_2. The distance (D_2) for full lists is the "bubble sort" distance, i.e., the number of pair wise adjacent transpositions needed to transform from one list to the other. The distance (D_2) between two lists of length n can be computed in n log n time using simple data structures. The above measures are metrics and extend in a natural way to several lists. Given several full lists L, M_1, ..., M_k, for instance, the normalized distance (D_1) of L to M_1, ..., M_k is given by-

D_1 (L, M_1, ..., M_k) = $\sum_i D_1(L, M_i)$ (3)

One can define generalizations of these distance measures to partial lists. If M_1, ..., M_k are partial lists, let U denote the union of elements in M_1, ..., M_k, and let be a full list with respect to U.

(3) Given one full list and a partial list, the distance (D1) weights contributions of elements based on the length of the lists they are present in. More formally, if L is a full list and M is a partial list, then -

SD_1 (L, M) = Σi D_1 M $|(M_i/|L|) - (M_i/|M|)|$ (4)

We will normalize SD_1 by dividing by |M|/2.

5.1 PROPOSED ALGORITHM

In our proposed algorithm, the distances are used to rank the various results. Let P_1, ..., P_n be partial lists obtained from various search engines. Let their union be S. A weighted bipartite graph for distance (D_1) optimization (N, SP, D_1) is defined as-

N = set of nodes to be ranked

SP = set of positions available

D_1 (e, p) = is the distance (from the Pi's) of a ranking that places element 'e' at position 'p', given by-

D_1 (e, p) = Σi | Pi(e)/|Pi| - p/n| (5)

where n = number of results to be ranked and |Pi| gives the cardinality of Pi.

Computation of aggregation for partial lists is NP-hard. Hence we have used distance measure (D_1). This problem can be converted to a minimum cost perfect matching in bipartite graphs. There are various algorithms for finding the minimum cost perfect matching in bipartite graphs.

Our proposed algorithm works as follows–

Step1: Calculate the reduced cost matrix from the given cost matrix by subtracting the minimum of each row and each column from all the other elements of it.

Step2: Cover all the zeroes with the minimum number of horizontal and vertical lines.

Step3: If the number of lines equals the size of the matrix, find the result.

Step4: If all of the zeroes are covered with fewer lines than the size of the matrix, find the minimum number that is uncovered.

Step5: Subtract it from all uncovered values and add it to any value(s) at the intersections of the lines.

Step6: Repeat until result is obtained.

5.2 MODEL COMPARISON

In evaluating the performance of the ranking strategies for all the queries, we have chosen precision as a good measure of relative performance because all the ranking strategies work on the same set of results and try to get the most relevant ones to the top.

Hence, a strategy that has a higher precision at the top can be rated better from the user's perspective. We have plotted the precision of the ranking strategies with respect to the recall. The recall is calculated as the number of relevant documents retrieved/total number of relevant results thus judged. It can be observed that on an average, our proposed ranking aggregation method gives better precision for the given set of results.

TABLE.1 : PRECISION OF SEVERAL RANK AGGREGATION METHODS AT A GIVEN RECALL.

CONDORCENT							
RECALL(R)	0.8	0.7	0.8	0.6	0.4	0.5	0.3
PRECISION (P)	0.12	0.24	0.48	0.6	0.72	0.84	0.96
BORDA							
RECALL (R)	0.9	0.5	0.5	0.6	0.4	0.5	0.3
PRECISION (P)	0.12	0.24	0.48	0.6	0.72	0.84	0.96
MY APPROACH							
RECALL (R)	0.9	0.6	0.9	0.8	0.6	0.6	0.3
PRECESION (P)	0.12	0.24	0.48	0.6	0.72	0.84	0.96

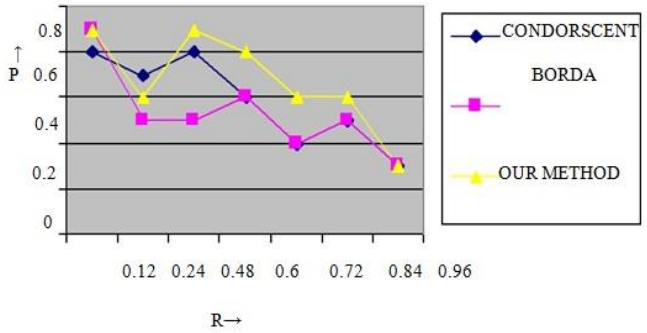

FIGURE 5.1 GRAPHICAL REPRESENTATION OF PRECISION AND RECAL

VI. RESULT

In evaluating the performance of the ranking strategies for the queries, we have chosen precision as a good measure of relative performance because all the ranking strategies work on the same set of results and try to get the most relevant ones to the top. A strategy that has a higher precision at the top can be rated better from the user's perspective. We have plotted the precision of the ranking strategies with respect to the recall.

We have proposed a rank aggregation method which works on our designed algorithm. This method has the advantage of being applicable in a variety of contexts and tries to use as much information as available. Our method is simple for implementation and do not have any computational overhead as compared to other methods. It is efficient, effective and provides robustness of search in the context of web.

In evaluating the performance of the ranking strategies for all the queries, we have chosen precision as a good measure of relative performance because all

the ranking strategies work on the same set of results and try to get the most relevant ones to the top.

A strategy that has a higher precision at the top can be rated better from the user's perspective. We have plotted the precision of the ranking strategies with respect to the recall. The recall is calculated as the number of relevant documents retrieved/total number of relevant results thus judged. It can be observed that on an average, our proposed ranking aggregation method gives better precision for the given set of results.

VII. REFERENCES

[1]. C.D. Manning, P. Raghavan, and H. Schutze. Introduction to Information Retrieval, February 2008. Draft, Cambridge University Press.

[2]. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. Modern Information Retrieval. ACM Press /Addison-Wesley,1999.

[3]. Bush, V. (1945). As We May Think. Atlantic Monthly. Vol. 176(1), 101-108.

[4]. Cleverdon, C.W. (1967). The Cranfield Tests on Indexing Language Devices. Aslib Proceedings, 19, 173-92.

[5]. Salton, G. (1968). Automatic Information Organisation and Retrieval. New York: McGraw-Hill.

[6]. (Baeza-Yates, R. & Ribeiro-Neto, B. (1999). Modern Information Retrieval. Reading, MA., Addison-Wesley.

[7]. Turtle, H. & Croft, W.B. (1990). Inference Networks for Information Retrieval. In Proceedings of the 13th Annual International ACM SIGIR Conference, Brussels, Belgium, pp. 1-24.

[8]. Robertson, S.E. & Sparck Jones, K. (1976). Relevance Weighting of Search Terms. Journal of the American Society for Information Sciences. 27(3): 129-46.

[9]. Salton, G. (1971). The SMART Retrieval System. Englewood Cliffs, NJ: Prentice Hall.

[10]. Harman, D. (1993). Overview of the First TREC Conference. Proceedings of ACM-SIGIR-93 Conference. Pittsburgh, PA. 36-47.

[11]. Fellbaum, C. (Ed). (1998). WordNet: An Electronic Lexical Database. Cambridge, MA. MIT Press.)

[12]. Baeza-Yates, R. & Ribeiro-Neto, B. (1999). Modern Information Retrieval. ACM Press, ISBN: 0-201-39829-X.

[13]. Hiemstra, D. (2000). Using Language Models for Information Retrieval. Enschede, The Netherlands, Neslia Paniculata.

[14]. Liddy, E.D. (1998). Enhanced Text Retrieval Using Natural Language Processing. Bulletin of the American Society for Information Science. Vol 24, No. 4.

[15]. Liddy, E.D., Paik, W., McKenna, M. & Yu, E.S. (1995). A natural language text retrieval system with relevance feedback. Proceedings of the 16th National Online Meeting.

[16]. Liu, X. & Croft, W.B. (2004). Statistical Language Modeling for Information Retrieval. In Cronin, B. (Ed.). Annual Review of Information Science & Technology. Vol.

[17]. Ponte, J. & Croft, W.B. (1998). A Language Modeling Approach to Information Retrieval. In Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval.

[18]. Robertson, S.E., Walker, S. & Beaulieu, M. (1998). Okapi at TREC-7. In Seventh Text REtrieval Conference (TREC-7), Gaithersburg, MD.

[19]. Salton, G., Fox, E. & Wu, H. (1983). Extended Boolean Information Retrieval. Communications of the ACM. 26(11). 1022-36.

[20]. Sparck Jones, K. & Willett, P. (Eds). (1997). Readings in Information Retrieval. San Francisco, Morgan Kaufmann Publishers.

[21]. Strzalkowski, T., Tzukermann, E. & Klavans, J. (2002). Information Retrieval and Natural Language Processing. In Mitkov, R. (Ed.),

Handbook of Computational Linguistics, Oxford University Press.

[22]. Estivill-Castro, V. and Yang, J. A Fast and robust general purpose clustering algorithm. Pacific Rim International Conference on Artificial Intelligence, pp. 208-218, 2000.

[23]. Fraley C. and Raftery A.E., "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis", Technical Report No. 329. Department of Statistics University of Washington, 1998.

[24]. Dhillon I. and Modha D., Concept Decomposition for Large Sparse Text Data Using Clustering. Machine Learning. 42, pp.143-175. (2001).

[25]. Fisher, D., 1987, Knowledge acquisition via incremental conceptual clustering, in machine learning 2, pp. 139-172.

[26]. Dempster A.P., Laird N.M., and Rubin D.B., Maximum likelihood from incomplete data using the EM algorithm. Journal of the Royal Statistical Society, 39(B), 1977.

[27]. Huang, Z., Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, 2(3), 1998.

[28]. Jain, A.K. Murty, M.N. and Flynn, P.J. Data Clustering: A Survey. ACM Computing Surveys, Vol. 31, No. 3, September 1999.

[29]. Zahn, C. T., Graph-theoretical methods for detecting and describing gestalt clusters. IEEE trans. Comput. C-20 (Apr.), 68-86, 1971.

[30]. Han, J. and Kamber, M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.

[31]. Banfield J. D. and Raftery A. E. . Model-based Gaussian and non-Gaussian clustering. Biometrics, 49:803-821, 1993.

[32]. J. I. Marden. Analyzing and Modeling Rank Data. Monographs on Statistics and Applied Probability, No 64, Chapman & Hall, 1995.

[33]. Meng, W., Yu, C., & Liu, K.-L., Building efficient and effective metasearch engines. ACMComputing Surveys, 2001, 34(1), 48–89.

[34]. Aslam, J. A., Montague, M., Models for metasearch. In: Proceedings of the 24th ACMSIGIR conference (pp. 276– 284), 2001.

[35]. Cynthia Dwork, Ravi Kumar, Moni Naor, D Siva Kumar, Rank Aggregation Methods for the web. In proceedings of the Tenth World Wide Web Conference, 2001.

[36]. Baeza-Yates, R., & Ribeiro-Neto, B., Modern information retrieval. New York: ACM Press, 2010.

[37]. Amitay, E., Carmel, D., Lempel, R., & So.er, A., Scaling IR-system evaluation using term relevance sets. In Proceedings of the 27th ACMSIGIR conference, 2004, pp. 10–17.

[38]. Soboro., I., Nicholas, C., & Cahan, P. Ranking retrieval systems without relevance judgments. In Proceedings of the 24th ACM SIGIR conference, 2001, pp. 66–73.

[39]. Croft, W. B., Combining approaches to information retrieval. In W. B. Croft (Ed.), Advances in information retrieval: recent research from the center for intelligent information retrieval. Kluwer Academic Publishers, 2000.

[40]. Cynthia Dwork, Ravi Kumar, Moni Naor, D Siva Kumar, Rank Aggregation Methods for the web. In proceedings of the Tenth World Wide Web Conference, 2010.

[41]. Fan, W., Fox, E. A., Pathak, P., & Wu, H. The effects of fitness functions on generic programming-based ranking discovery for Web search. Journal of the American Society for Information Science and Technology, 55(7), 2004, 628–636.

[42]. Hawking, D., Craswel, N., Bailey, P., & Gri.ths, K., Measuring search engine quality. Information Retrieval, 4(1), 2001, 33–59.

[43]. Nuray, R., & Can, F., Automatic ranking of retrieval systems in imperfect environments. In

Proceedings of the 26th ACM SIGIR conference 2009, pp. 379–380.

[44]. S. Brin and l. page, "The anatomy of a large-scale hypertextual web search engine", in proceedings of the 7th world wide web conference: pp. 107-117, 199

[45]. Adi Wahyu Pribadi, Zaenal Arifin Hasibuan, 2003, “Implementing Inference Network For Information Retrieval System In Indonesian Language”. Conference: iiWAS'200

Cite this article as :

Abhay Dwivedi, Ankit Maurya, Sandhya Rawat, "Improving Information Retrieval Performance", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 5, pp. 52-63, September-October 2022.
Journal URL : https://ijsrcseit.com/CSEIT228515