Comparative Analysis of Intersection Algorithms on Queries using Precision, Recall and F-Score

Sachin Kumar
Sr.Asst.Prof – Computer Science/IT
Fairfield Institute of Management & Technology
(Affiliated to GGSIP University, New Delhi)

Pratishtha Gupta, PhD
Asstt. Professor – Department of Computer Science
Banasthali Vidyapith,Jaipur,India

ABSTRACT
Search Engines are the software systems to search for the required information on the World Wide Web. Though Search Engine Optimization deals with increasing the visibility of Web pages on search results, this research work focuses on the search algorithm, which is expected to bring the most appropriate, relevant and required documents for the given search query. The demand of the new such algorithm has been presented by a comparative study of intersection algorithm and its variants. CRANTOP database is a standard database available on WWW, which not only provides the database of documents and queries but also provide the actual relevance of documents for the corresponding queries. Study has been done on a sample part of that database, which is available in appendix and analysis has been done on the basis of Precision, Recall and F-Measure values.

Keywords
Precision, Recall, F-measure, Algorithms, Database et al.

1. INTRODUCTION
Algorithm used for Information Retrieval impacts the customer satisfaction a lot. Many a times, lot much data required by the user is available but cannot be retrieved just because the algorithm used could not understand the requirement of the user. This satisfaction of the user requirements is measured in many ways. Precision, Recall and F-Score are some of those measurement models, through which effectiveness of an algorithm can be measured.

This paper emphasizes on the evaluation of a query from a standard database CRANTOP, using a simple intersection algorithm and its variants. Comparisons in the results have been done to emphasize the fact that differences exist in the outcomes of the same query on the same database.

2. REVIEW OF LITERATURE
According to a study conducted by [3] included theoretical, methodological and empirical aspects to explore modalities of use and suggest further avenues. However [4] feels that the objective of this paper is to characterize the changes in the rankings of the top ten results of major search engines over time and to compare the rankings between these engines. [2] suggested in his studies that term position information, as provided in some Boolean systems in the form of field restriction and term proximity, is reviewed and its value assessed. Non-Boolean retrieval in the form of the ranked output experiment has not so far used term position information but has concentrated on schemes of term weighting.

[1] revealed that in the first approach we set the same threshold for each of the IF systems. In the second approach the threshold of each IF system is tuned independently to maximise its own EP ("local optimisation"). In the third approach the thresholds of the IF systems are jointly tuned to maximise the EP of the combined system ("global optimisation"). [7] found that an interdisciplinary framework for list, especially information retrieval (IR), in a way that goes beyond the cognitivist 'information processing paradigm'. The main problem of this paradigm is that its concept of information and language does not deal in a systematic way with how social and cultural dynamics set the contexts that determine the meaning of those signs and words that are the basic tools for the organisation and retrieving of documents in List.

[6] founded that the objective of the paper is to amalgamate theories of text retrieval from various research traditions into a cognitive theory for information retrieval interaction. Set in a cognitive framework, the paper outlines the concept of polyrepresentation applied to both the user's cognitive space and the information space of IR systems. The concept seeks to represent the current user's information need, problem state, and domain work task or interest in a structure of causality. [8] said that the purpose of this article is to review the research on human-computer interfaces for library-based commercial online information retrieval (IR) systems.[10] suggested that when developing an IR system it is necessary to think of it as a controllable system. It makes the system capable of fulfilling this system. It leads to its requirements for an optimal search and optimization is one of the problem of control.

[9] feels that the purpose of this paper was to study users' behaviour when using different search engine results pages (SERPs) to identify what types of scents (cues) were the most useful to find relevant information to complete tasks on the Web based on information foraging theory. [5] described the design and implementation of a system for computer generation of linked HTML documents to support information retrieval and hypertext applications on the World Wide Web. The approach is based on work by Salton and others, but extends the concept to be compatible with the World Wide Web browser environment by adding an interactive indexing technique that is well suited to the mouse-based point-and-shoot input common to windowed browsers. The system does not require text query input, nor any client or host processing other than hypertext linkage.

The goal of this work is to construct a fully automatic system in which original text documents are read and processed by a computer program that generates HTML files, which can be used immediately by Web browsers to search and retrieve the original documents. Thus, a user with a large collection of information — for instance, newspaper articles — can feed these documents to the program described here and produce directly, without further human intervention, the necessary files to establish World Wide Web home and related pages, to

support interactive retrieval and distribution of the original documents.

2.1 Intersection Algorithm on Information Retrieval

The intersection operation is the crucial one; it needs to efficiently intersect postings lists so as to be able to quickly find documents that contain both terms. There is a simple and effective method of intersecting postings lists using the intersection algorithm. It maintains pointers into every lists and walk through the every postings list simultaneously, in time linear in the total number of postings entries. At each step, it compares the docID pointed to by every pointer. If they are the same, put that docID in the results list, and advance every pointers. Otherwise, advance the pointers pointing to the smaller docID. If the lengths of the postings lists are x and y , the intersection takes O(x+y) operations. Formally, the complexity of querying is O(N), where N is the number of documents in the collection.Its indexing methods gain it just a constant, not a difference in O time complexity compared to a linear scan, but in practice the constant is huge. To use this algorithm, it is crucial that postings be sorted by a single global ordering.Using a numeric sort by docID is one simple way to achieve this.This algorithm can be extended to process more complicated queries like for example.

(Brutus OR Caesar) AND NOT Calpurnia
Query optimization is the process of selecting how to organize the work of answering a query so that the least total amount of work needs to be done by the system. A major element of this for Boolean queries is the order in which postings lists are accessed. For each of the t terms, it needs to get its postings, then AND them together. The standard heuristic is to process terms in order of increasing document frequency, if it start by intersecting the two smallest postings lists, then all intermediate results must be no bigger than the smallest postings list, and we are therefore likely to do the least amount of total work, Christopher D. M. et al[2005].

It keeps the frequency of terms in the dictionary it allows to make this ordering decision based on in-memory data before accessing any postings list. It will get the frequencies for all terms, and it can then estimate the size of each OR by the sum of frequencies of its disjuncts.

2.2 Intersection via Skip Pointers Algorithm on Information Retrieval

Here also like intersection extensions to postings list data structures and ways to increase the efficiency of using postings lists. It recalls the basic postings list intersection operation if it walks through two or more postings lists simultaneously, in time linear in the total number of postings entries. If the list lengths are m and n, the intersection takes O(m+n) operations. Can it be better than this? That is empirically, it can usually process postings list intersection in sub linear time. It can be possible if the index is not changing too fast, Christopher D. M. et al [2005].
One way to do this is to use a skip list by augmenting postings lists with skip pointers at indexing time. Skip pointers are effectively shortcuts that allow to avoid processing parts of the postings list that will not figure in the search results. The two questions are arises here that where to place skip pointers and how to do efficient merging using skip pointers. Here a number of variant versions of postings list intersection with skip pointers is possible depending on when exactly it checks the skip pointers. Its results is again similar to the intersection algorithm kind of results as shown below in query 1 to 50 and the recall and precision parameters values are same.

2.3 Positional Intersection Algorithm on Information Retrieval

Here, for each term in the vocabulary, it stores postings of the form docID:(position1,position2,…..), where each position is a token index in the document. Each posting will also usually record the term frequency. To process a phrase query, it still needs to access the inverted index entries for each distinct term. Here it finds the results where there are docID on the same position in the document in every postings list in a query that is this query will be retrieved otherwise it cannot be retrieved if the same positions are not found in the postings list of any of the queries in the database, Christopher D. M. et al.[2005].

3. PROPOSED ALGORITHM STEPS

1.) Break Query :- Break-row queries are often loosely referred to as terms or words, but it is sometimes important to make a type distinction.

2.) Phrase Queries: - A phrase is a collection of words in sentence in a document like “A cakes and bakers manufacturing company has many distributors” is not a match. Most recent search engines support a double quotes syntax (“manufacturing company”) for phrase queries.

3.) Two –Words Phrases :- The two words query is the two- words phrases in that one approach to handling phrases is to consider every pair of consecutive terms in a document as a phrase.

4.) Three –Words Phrases:- The three words query is the three- words phrases in that it has increased the priority of a root words in the database to be retrieved break them a phrase in single root words so that they can retrieve accurate amount of information from the database.

5.) Remove Stopwords :- Common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. These words are called stop words.

6.) Convert Remaining Into Root Words :- Here may be a root words like representation, represented or representing. It will make the accurate data to be retrieved from the set of database as a root word.

7.) Classify Into Specific And General Words :- General words and specific words are not opposites instead, they are the different ends of a range of words. General words refer to groups whereas specific words refer to individuals.

8.) Synonyms :- Similar meaning root words are called synonyms

9.) Intersection With Specific Words :- Intersection have taken as the common postings together as if it have two Boolean queries located in the dictionary. Then intersection operation need to efficiently intersect postings lists so as to be able to quickly find documents that contain both terms.

4. CRANTOP DATABASE

Crantop database is an xml based database in which this research is carried out. It contains 3 parts that is documents with their Doc IDs in their dictionaries. This database would contains Doc.ID 1 to Doc.ID 1400 in which there are several nodes of linked list structure which can be found by the help of softwares like MS-Word. In the second database it contains different queries i.e. the research would be carried out on query 1 to 50 its analysis would be carried out in MS-Excel by sorting the every words into ascending order and make it into a new sparse matrix representation. In the third database its every documents ranking would be given that is how many times a single link list can be carried out in a single query from 1 to 50 for the Doc ID's 1 to 1400 and its rankings would be given here on the basis of that its precision and recall can be calculated here with its f-measure with different Intersection, Intersection with skip pointers and positional intersection algorithms and based on that new algorithm would be proposed and based on that second table would be analysed. This is the need of this crantop database by this research can be carried out.This database is available at the link given below **ftp://ftp.cs.cornell.edu/pub/smart/cran/**

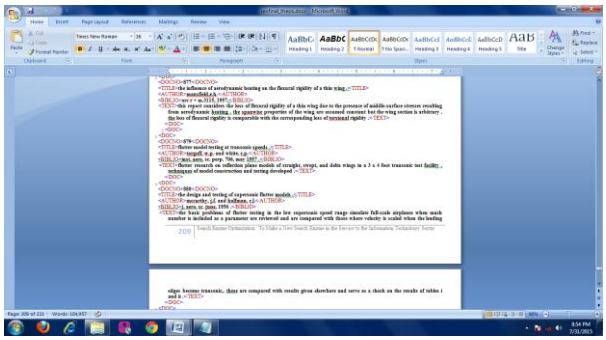

Fig1: Snapshot of Documents of CRANTOP database

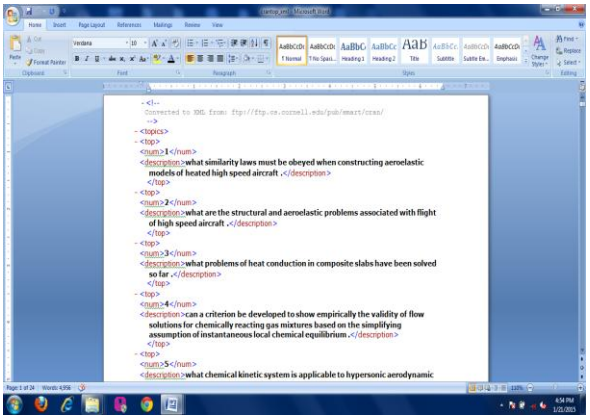

Fig2: Snapshot of Queries of CRANTOP database

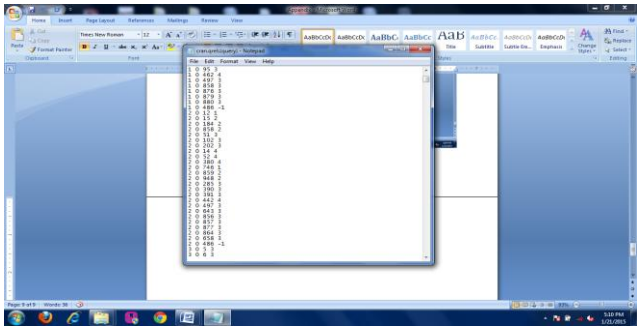

Fig3: Snapshot of relevance result of CRANTOP database

5. PRECISION AND RECALL PARAMETERS

The two most frequent and basic measures for information retrieval effectiveness are precision and recall. Information retrieval system returns a set of documents for a query.

Precision (P) is the fraction of retrieved documents that are relevant

Precision= (relevant items retrieved) / (retrieved items)

Recall (R) is the fraction of relevant documents that are retrieved

Recall= (relevant items retrieved) / (relevant items)

Typically web surfers would like every result on the first page to be relevant (high precision) on the other hand professional searchers such as paralegals and intelligence analysts are very concerned with trying to get as high recall as possible, and will tolerate fairly low precision results in order to get it, Christopher D. M. et al[2005].

A single measure that trades off precision versus recall is the F measure, which is the weighted harmonic mean of precision and recall.

F- measure = 2 * Precision* Recall / Precision + Recall

The advantage of having the two numbers for precision and recall is that one is more important than the other in many circumstances .Typically, web surfers would like every result on the first page to be relevant (high precision) but have not the slightest interest in knowing let alone looking at every document that is relevant. In contrast, various professionals searchers such as paralegals and intelligence analysts are very concerned with trying to get as high recall as possible, and will tolerate fairly low precision results in order to get it. Recall is a non-decreasing function of the number of documents retrieved whereas precision usually decreases as the number of documents retrieved is increased. Then here a single measure that trades off precision versus recall is the F-measure, which is the weighted harmonic mean of precision and recall Christopher D. M. et al[2005].

To get more high precision and recall we use the proposed algorithm which I will be discussed here step wise and it will show the difference between these three algorithms which are discussed here and the proposed one. The proposed algorithm stepwise to be discussed above.

There is a good reason why we use precision, recall and f-measure to measure the proposed algorithm for information retrieval problems. In almost all circumstances, the data is extremely skewed : normally over 99.9% of the documents are in the nonrelevant category. A system tuned to maximize accuracy can appear to perform well by simply deeming all documents nonrelevant to all queries. Even if the system is quite good trying to label some documents as relevant will almost always lead to a high rate of false positives. Users are always going to want to see some documents and can be assumed to have a certain tolerance for seeing some false positives providing that they get some useful information. The measures of precision and recall here concentrate the evaluation on the return of true positives, asking what percentage of the relevant documents have been found and how many false positives have also been returned. Again, there will be a conclusion that if we measure it on CRANTOP database with intersection algorithms and its alias its precision, recall and f-score are less as compared to the

proposed algorithm will make it is high to all the parameters like precision, recall and f-score the comparisons are given in below two tables.

6. MEASUREMENT OF PRECISION AND RECALL AT START OF THE COMPARATIVE ANALYSIS WITH F-MEASURE

It is analyzing that table on intersection algorithm and its another alias on 1400 documents and 50 queries. They are finding it through CRANTOP database which are relevant documents and how many documents are retrieved from the total relevant documents from the total documents found and now finding the Precision, Recall and F-score which are very low as compared to the proposed algorithm.

Table1. Measurement of Precision & Recall with F-measure

Q.NO	Intersection algorithm and its alias				Precision	Recall	F-score
	Relevant doc	Doc retrieved	Relevant doc.	Total doc found	Result	Result	Result
1	0	0	29	29	0	0	0
2	0	0	25	25	0	0	0
3	0	0	9	9	0	0	0
4	0	0	3	3	0	0	0
5	0	1	4	5	0	.25	0
6	2	11	9	5	.182	2.2	0.34
7	0	0	6	6	0	0	0
8	0	1	5	5	0	.2	0
9	0	0	4	4	0	0	0
10	0	0	9	9	0	0	0
11	1	0	8	8	0	.125	0
12	1	0	7	7	0	.143	0
13	0	0	5	5	0	0	0
14	2	8	3	3	.25	2.67	0.46
15	1	0	3	3	0	.333	0
16	1	0	4	4	0	.25	0
17	1	0	3	3	0	.333	0
18	1	0	4	4	0	.25	0
19	0	0	10	10	0	0	0
20	0	0	10	10	0	0	0
21	1	0	5	5	0	.2	0
22	1	0	2	2	0	.5	0
23	0	0	33	33	0	0	0
24	1	0	4	4	0	.25	0
25	0	0	10	10	0	0	0
26	0	0	7	7	0	0	0
27	0	1	3	3	0	.333	0
28	0	1	3	3	0	.333	0
29	1	2	10	10	.5	.3	0.38
30	0	0	8	8	0	0	0
31	0	0	2	2	0	0	0
32	1	1	7	7	1	.286	0.44
33	3	2	4	4	1.5	1.25	1.36
34	0	3	7	7	0	0	0
35	0	0	4	4	0	0	0
36	0	1	3	3	0	.333	0
37	0	1	10	10	0	.1	0
38	0	0	11	11	0	0	0
39	1	0	14	14	0	.07	0
40	0	1	12	12	0	.083	0
41	2	0	4	4	0	.5	0
42	0	0	6	6	0	0	0
43	0	0	7	7	0	0	0
44	0	0	4	4	0	0	0
45	1	5	13	13	.2	.43	0.27
46	0	1	16	16	0	.0625	0
47	0	1	15	15	0	.066	0
48	0	0	12	12	0	0	0
49	0	1	3	3	0	.333	0
50	0	0	7	7	0	0	0

7. COMPARATIVE ANALYSIS OF PRECISION AND RECALL AFTER APPLYING PROPOSED ALGORITHMS AND FIND F-MEASURE

Here again it is analyzing the 1400 documents and on 50 queries of CRANTOP database. It is again analyzed on the proposed algorithm i.e. results of proposed algorithms which are found retrieved and also the relevant results in that how many are matched and how many are relevant results and on proposed algorithm how many are relevant results then we found the precision, recall and f-score now we were found that its values of precision, recall and f-score will be increased here after analyzed.

Table2. Demonstration of proposed algorithm through Precision, Recall and F-measure

Q.no	Results of propose algo.	Relevant result	Match ID's	Relevant result	Propose Algo Relevant Results	Precision	Recall	F-score
1	12,13,24,29,47,51,141,172,184,431,497,519,649,700,746,798,876,879,880,948,976,1197,1289,1379	12,13,14,15,31,57,66,95,462,497,858,879,880,184,29,51,102,378,859,185,30,37,52,142,195,875,56,876,486	**9**	29	25	0.36	0.310345	1.08
2	12,14,24,29,47,51,141,172,184,431,497,519,649,700,746,798,876,879,88	12,14,15,51,102,202,380,390,391,658,877,948,184,52,380,746,859,94	**7**	25	25	0.28	0.28	0.84

	0,948,976,1197,1289,1379	8,285,390,391,442,643,858,497,856,857,877,864,658,486,858,643						
3	5,6,90,91,144,168,181,399,485,518	5,6,181,144,485,90,91,119,399	**7**	9	10	0.7	0.7778	2.1
4	236,401,1296,1297	166,236,488	**1**	3	4	0.25	0.3333	0.75
5	19,28,31,37,48,56,57,68,85,95,122,160,272,304,305,307,308,310,360,401,495,497,525,536,544,552,556,557,570,572,976,1076,1272,1379	401,488,1297,522,1296	**1**	5	34	0.029412	0.2	0.0882352
6	257,418,491,558,976	99,115,257,258,491	**2**	5	5	0.4	0.4	1.2
7	48,56,57,58,354,469,492,717	19,20,56,57,58,492	**4**	6	8	0.5	0.6667	1.5
8	48,56,57,58,354,469,492,717	20,999,1005,48,122,58,196,354,360,197,1112,492	**4**	5	8	0.5	0.8	1.5
9	21,22,24,37,11	22,534,21,550	**4**	4	19	0.210526	1	0.63
	9,144,269,270,283,306,354,378,436,485,534,550,554,623,1226							1579
10	166,167,168,169,185,236,405,427,488,493,518	259,405,302,436,437,438,908,1011,493	**2**	**9**	11	0.181818	0.2222	0.545455
11	20,27,28	27,28,262,160,20,263,654,495	**3**	**8**	**3**	1	0.375	3
12	1,14,29,51,66,141,142,172,185,272,289,486,658,704,715,749,859,877,899,999,1197,1272,1289,1333	86,194,650,649,652,624	**0**	**6**	24	0	0	0
13	199,252,315,439,440,467,468,469,496,503,521,526,643,799,800,879,903,919,1290	64,265,65,311,496	1	**5**	19	0.052632	0.2	0.1578995
14	64,65,263,265,517,609,903	64,65,496	**2**	**3**	**7**	0.285714	0.6667	0.857143
15	30,184,195,462,463	463,462,497	**2**	**3**	**5**	0.4	0.6667	1.2
16	2,3,4,3	266,10	**1**	**4**	**8**	0.12	0	0.

	9,86,94,106,418	6,196,498				5	.25	375
17	8,15,94,106,199,242,259,279,326,700,714,799,903,1112	106,196,498	**1**	**3**	14	0.071429	0.333333	0.2142 86
18	52,56,57,58,85,121,122,173,196,197,215,247,250,259,354,360,409,469,492,498,514,528,553,717,1259	196,197,198,498	**3**	**4**	25	0.12	0.75	0.36
19	142,164,555,1379	32,67,164,639,715,716,719,1379,717,499	**2**	**10**	**4**	0.5	0.2	1.5
20	87,88,268,269,270,500	87,88,104,267,268,269,270,407,408,500	**6**	**10**	**6**	1	0.6	3
21	68,167,185,236,283,302,405,427,524,1355	271,16,413,414,502	**0**	**5**	10	0	0	0
22	2,9,17,23,132,145,493	68,502	**0**	**2**	**7**	0	0	0
23	1,14,29,141,142,185,859,877,899,1112,1197,1272,1289,1333	900,902,200,201,601,899,903,593,199,594,901,544,597,749,917,919,1333,634,687,698,1290,700,704,705,1109,1112,1141,1197,	**5**	**33**	14	0.357143	0.151515	1.07 1429
		1256,1259,1272,1289,892						
24	12,51,756	46,47,92,756	**1**	**4**	**3**	0.333333	0.25	1
25	7,31,48,52,95,121,122,173,187,188,200,201,212,213,214,216,224,225,242,259,272,276,278,279,391,406,409,428,464,511,513,519,880,1197,1259,1272	213,212,214,215,216,276,277,426,427,511	**6**	**10**	36	0.166667	0.6	0.5
26	3,4,8,11,16,79,94,104,145,207,265,271,306,307,316,326,481,493,526,527,528,565,570,572,611,629,798,1076,1109,1185,1226,1355	145,611,376,406,565,1076,511	**4**	**7**	32	0.125	0.571429	0.375
27	1,30,195,199,201,225,247,250,279,288,289,433,497,520,643,652,704,705,749,752,877,901,919,1289,1290,1333	224,278,428,512	**0**	**4**	26	0	0	0
28	168,247,470,5	224,279,512	**0**	**3**	**4**	0	0	0

	93							
29	55,251,464,465,466,601,612,752,879,901,902,919,1289	250,514,609,225,793,464,465,612,466,513	**4**	**10**	13	0.307692	0.4	0.923077
30	123,250,289,464,465,466,513,556,557,572,752,902,919,1197	225,464,514,466,609,612,793,513	**3**	**8**	14	0.214286	0.375	0.642857
31	173,199,215,247,277,433,487,512,597,652,698,749,751,775,903,919,1076	776,751	**1**	**2**	17	0.058824	0.5	0.176471
32	4,250,302,306,611,752,777,877,1355	465,249,777,778,247,250,752	**3**	**7**	**9**	0.333333	0.428571	1
33	431,466,486,514,516,544,553,594,609,612,672,672,714,719,799,800,876	252,431,141,516	**2**	**4**	17	0.117647	0.5	0.352941
34	431,466,486,514,516,544,553,594,609,612,672,714,719,799,800,876	252,431,672,714,799,800,516	**6**	**7**	16	0.375	0.857143	1.125
35	68,166,167,169,185,405,488	166,167,132,517	**2**	**4**	**7**	0.285714	0.5	0.857143
36	13,21,29,51,66,119,142,269,2	169,168,518	**0**	**3**	22	0	0	0
	70,283,354,436,437,438,485,493,524,550,554,623,1185,1226							
37	97,121,173,188,242,277,409,427,431,465,487,519	173,188,172,97,121,187,242,409,487,519	**8**	**10**	12	0.666667	0.8	2
38	7,8,9,19,28,37,56,57,68,85,79,122,160,187,207,272,304,305,307,308,310,314,315,360,401,418,505,525,536,544,556,557,558,570,572,976,1076,1272,1379,	24,283,552,272,557,558,553,554,555,556,536	**5**	**11**	39	0.128205	0.454545	0.384615
39	2,3,4,7,8,9,12,16,17,21,22,23,37,79,94,104,145,207,265,271,306,307,314,316,326,376,406,481,493,526,527,528,565,570,572,611,629,798,899,1076,1109,1185,1226,1355	272,555,24,283,552,79,207,418,557,554,556,505,1257,536	**2**	**14**	44	0.045455	0.142857	0.136364
40	7,8,9,19,28,37,56,57,68,85,79,122,1	24,283,552,272,85,976,557,558,553,	**7**	**13**	39	0.179487	0.538	0.538462

	60,187,207,272,304,305,307,308,310,314,315,360,401,418,505,525,536,544,556,557,558,570,572,976,1076,1272,1379,	554,555,556,536					462	
41	289,433	288,289,433,520	**2**	**4**	**2**	1	0.5	3
42	194,200,216,439,440,467,468,469,496,503,521,526,797,919	468,467,469,470,775,521	**4**	**6**	14	0.285714	0.666667	0.857143
43	194,216,467,468,469,496,503,521,526,797,919	467,468,39,503,775,469,521	**5**	**7**	11	0.454545	0.714286	1.363636
44	102,401,552,746,1226,1296,1297	302,436,437,524	**0**	**4**	**7**	0	0	0
45	2,14,305,308,310,525,572,1185	305,570,308,481,338,1226,1355,1185,629,663,798,572,525	**5**	**13**	**8**	0.625	0.384615	1.875
46	9,17,19,28,37,84,122,123,160,304,305,338,344,360,481,495,525,544,572,623,1076,1185	305,338,344,481,84,123,1185,623,570,798,1226,629,663,572,1355,525	**10**	**16**	22	0.454545	0.625	1.363636
47	2,9,14,22,87,2	304,306,307,3	**9**	**15**	17	0.529412	0.	1.58
	07,304,305,306,308,309,310,464,525,572,611,663	05,629,663,308,309,310,570,798,1185,1355,572,525					6	8235
48	2,8,9,197,200,304,307,308,310,315,439,440,464,511,513,514,521,526,544,572,593,601,634,797,799,800,879,880,900,901	439,311,316,440,187,314,315,797,798,794,265,526	**5**	**12**	30	0.166667	0.416667	0.5
49	23,320,478,527	320,478,527	**3**	**3**	**4**	0.75	1	2.25
50	106,188,196,197,279,326,468,498,528,550,714,1112,1259	326,629,21,22,94,306,528	**2**	**7**	13	0.153846	0.285714	0.461538

8. CONCLUSION

Comparative Analysis of very simple intersection algorithms have been done on queries, which have been picked from the standard crantop database. More advanced algorithms are in use, but the simple algorithms have been picked to emphasize the need of a new algorithm by showing the computed values of Precision, Recall and F-Score. These values are very low, depicting the need to develop a novel idea for searching documents. After applying the advanced proposed algorithm the values of Precision, Recall and F-Score will become high as compared with previous algorithms for the information retrieval. The significance of new algorithm is that its database can take root words, Break Query, Phrase Queries, Remove Stopwords, Convert Remaining Into Root Words, Classify Into Specific And General Words, Synonyms, Intersection With Specific Words by doing so it can increase the value of Precision, Recall and F-Score if we are using sparse matrix structure to do that.

9. REFERENCES

[1] Alexander B. et al., 2009 "A decision theoretic approach to combining information filtering", Online Information Review, Vol. 33 Iss: 5, pp.920 – 942

[2] E. Michael Keen, 1991 "The use of term position devices in ranked output experiments", journal of documentation, vol. 47 iss: 1, pp.1 – 22

[3] François X. D. et al.,2014 "applying and theorizing institutional frameworks in is research: a systematic analysis from 1999 to 2009", information technology & people, vol. 27 iss: 3, pp.280 – 317

[4] Judit b. et al., 2006 "methods for evaluating dynamic changes in search engine rankings: a case study", journal of documentation, vol. 62 iss: 6, pp.708 – 729

[5] Kevin C. O., 1996 "world wide web-based information storage and retrieval", online and cd-rom review, vol. 20 iss: 1, pp.11 – 20

[6] Peter I., 1996 "cognitive perspectives of information retrieval interaction: elements of a cognitive ir theory", journal of documentation, vol. 52 iss: 1, pp.3 – 50

[7] Soren B., 1996 "Cybersemiotics: a new interdisciplinary development applied to the problems of knowledge organisation and document retrieval in information science", journal of documentation, vol. 52 iss: 3, pp.296 – 344

[8] S.M. Zabed Ahmed et al., 2009 "A review of research on human-computer interfaces for online information retrieval systems", The Electronic Library, Vol. 27 Iss: 1, pp.96 – 116

[9] Stella D. T. et al. ,2014 "Using cues to forage for information on the Web", Journal of Systems and Information Technology, Vol. 16 Iss: 4, pp.296 – 312

[10] Valery J. F. et al., 1997, Control and feedback in ir systems, in valery j. Frants, jacob shapiro, vladimir g. Voiskunskii (ed.) Automated information retrieval: theory and methods (library and information science, volume 97a) emerald group publishing limited, pp.222 – 259

[11] ConvertedtoXMLfrom:ftp://ftp.cs.cornell.edu/pub/smart/cran/ (for 1400 documents)

[12] ConvertedtoXMLfrom:ftp://ftp.cs.cornell.edu/pub/smart/cran/(for 50 queries)

[13] EllisHorowitz, Sartaj Sahni, Dinesh Mehta, Fundamentals of data structures in C++,2nd Edition ,2008, Universities Press(India) Private Limited.

[14] Christopher D. Manning(Stanford University) ,Prabhakar Raghvan(Yahoo! Research), Hinrich Schutze(University of Stuttgart) , Introduction to Information Retrieval,2009,Cambridge University Press.