# Collection and Selection Based Relevant Degrees Of Documents

Mechach Kheira[1, 3], Zekri Lougmiri[2, 3], Abdi Mustapha Kamel[1, 3]
[1]RIIR Laboratory
[2]LAPECI Laboratory
[3]Department of computer science
University of Oran1 Ahmed Ben Bela
BP 1524 El-M'naouer Maraval, Oran, Algéri
mechach.kheira@gmail.com, abdi.mustapha@univ-oran.dz, lougmiri@gmail.com

*Journal of Digital Information Management*

***ABSTRACT:*** *In this paper, we address the problem of selection collections. This is important for locating responses in digital libraries. The aim of methods, which deal with the area of information retrieval, is to reduce the amount of the exchanged messages by selecting the best servers from the beginning of the search. We propose a new function of selection based on statistics. Our function takes into account the relevance degree of documents in order to rank collections. We have implemented and compared our function with other methods. The experimentations have shown that our proposition is very competitive.*

**Categories and Subject Descriptors**
**H.3.3 [Information Search and Retrieval]**

**General Terms:**
Digital Libraries, Information Retrieval, Relevance

**Keywords:** Collection Selection, CORI, CSRD, CS, Relevance, Degree Of Relevance, Distributed System, Digital libraries

**Received:**
13 December 2014, Revised 18 January 2015, Accepted 20 January 2015

## 1. Introduction

Nowadays, search engines are ubiquitous since they are continuously consulted for supporting search in actual large databases. They are efficient in term of high quality of responses to the submitted queries. Indexes servers and databases servers are the principal components of any system. They cooperate in order to search the best results for users'queries. Indexes are large collections; they serve to locate in databases, with high probability, the relevant responses. In a point of view of size, these indexes are too large, may be this size can be larger than databases'one. Theoretically, an index is an inverted list which captures, for every term, the list of documents where it appears. Other information, like its occurrences, positions, type of the police, is also saved in this inverted list. By this composition, the size of the index is so large. As much information is saved, a potential response can be determined with high probability in a short time.

In order to locate information, many systems have been designed. They implement efficient mechanisms in order to select the best collections. First systems were centralized and a global centralized index has been implemented. This index served in the searching process. The inconvenient with this method is that these systems cannot scale with the actual growing information sources. In order to overcome this problem, the designers propose to distribute the storing and searching processes. This manner can lead to a simpler way to maintain and manage the local indexes. Typically, a distributed system is constructed of a set of servers and every one possesses a set of collections of documents. A central manager, called also broker, can interrogate these servers. When it receives a query, the broker plays the principal role. At first, it selects and questions a subset of servers (collections). Every selected server searches locally the potential collections and returns its response. At the second role,

the broker merges the responses in one list and sorts it. After that, it returns the list to the user.

Our work deals with the selection collection problem. We study the relevance of the selected collections. We give briefly, the theoretical aspects of CORI (Callan et al 1995) and CS for Collection Selection (Abbaci et al 2002) methods. These methods use some parameters which are obtained from heuristics. These parameters can change from a collection to another (D'Souza et al 2004). The changes make these methods unstable. We present a new and simple method for selecting collections. We do not use any extra-collection parameter. The goal of this method is to give the highest score to the collection which contains the most relevant documents. At the difference of CORI, we do not give any endowment to the collections with the highest number of documents concerned by one term. The relevance of a collection depends on the documents relevance. A document is relevant if it shares terms with the query. Thus, we give more importance to the quality of responses than to their number. In order to touch to our assumption, we compare our method to CORI and CS.

In reality, we do not like extra-collection parameters because they bias the search. As we will show bellow, it is not easy to find the best values to parameters; for every dataset, designers must calibrate their values in order to make methods stable. Doubtless, the magic success of PageRank (Page et al 1999) resides in the logical and fair criterion for ordering pages. PageRank's authors choose the popularity degree of pages which is presented by the in-degree of pages in the directed web graph. Such criterion is independent from user's query and from the referenced documents in Google.

In our case, we have chosen to study the nature of collections and then to find a non-parameterized formula. A good function must be deduced from the anatomy of the collections. For every query, we compute at the moment of the questioning process the different sets of documents in the collections. This calculus produces different hidden clusters without outputting them. The sets of eligible documents are deduced according to our definition of the relevance of a document and a collection to the query.

The remainder of this paper is structured as follows. Section 2 presents some related works. In section 3, we present the CORI method. It is the state-of-the-art and is inevitable in such work. This section presents also the CS method. We give the advantages and the inconvenients of them. Section 4 contains our approach. We begin by giving some definitions related to our function. We define the relevance and the relevance degrees of documents and collections in the first part of this section. These definitions are at the basis of our function. In the second half of section 4, we give the details of our function of collection selection. Section 5 presents the experimentations results. We use the dataset Reuters-RCV1(Lewis et al 2004, Norvag et al 2008). Finally in section 6, we give a conclusion and future works.

## 2. Related Works

Collection selection problem has received a big attention. The most important challenge is to reach the smallest number of servers but returning the best answers. In this field, we find methods based on terms dictionary like (Callan et al 1995), (Craswell et al 2000), (French et al 1999), (Meng et al 2001). In these systems, a central server detains a copy of the dictionary of all terms of all collections. CORI is perhaps the most known method in this context; it presents the state-of-the-art. Other methods exist and are based on statistical approach on TF*IDF (Spaärk 1972). Works like (Sergey et al 2007) and (Jie et al 2004) are based on model language and pseudo-relevance. (Jie et al 2006) extended user modeling for full-text federated search in peer-to-peer systems. The presented approach uses past queries in order to perform efficient searches for future queries. This work looks on terms which seem the most interesting for the user. Holz in (Holz et al 2007) has proposed a framework for evaluating semantic search in peer to peer systems and has incorporated CORI in the evaluation process. (Witshel et al 2008) have studied the problem of ranking information sources in peer to peer systems. Authors have used a modified form of the CORI's ranking function for merging results. Salampasis in (Salampasis et al 2013) has presented a system for classifying patents in order to select collections. They have integrated searching tools in there system based on that classification. The proposed method was compared to CORI and seems giving better results. (Khalaman et al 2012) have proven that similarity inter-document is an efficient tool for federating searches. Authors have proposed a function based on heuristics. This function evaluates queries on clusters of documents. The performances of this method have been compared with those produced by CORI.

## 3. Collection selection

Collection selection refers to the automatic or manual selection of a subset of collections that potentially contain relevant documents to a given query. The negligence of this step and the forwarding of the query to all available collections, at the querying moment, is an overwhelming operation. So, collection selection reduces the amount of messages exchanged in the system without reducing the efficiency of the searching method (Hawking et al 1999) (French et al 1999). Note that a searching method is efficient if it reduces the number of false positives and increases the number of false negatives. In the rest of this section, we review the CORI and CS methods.

### 3.1. The CORImethod

In a distributed environment for searching information (Callan 2000), we suppose a query $Q$, consisting of a set of terms $<t_1, t_2,..., t_m>$, and a set $C$ of $N$ collections. Every

collection $c \in C$ contains $f_c$ documents and is considered as a bag of words. In such environment, the CORI method is developed. It is a classification algorithm for the INQUERY system (Callan et a 1992). CORI deals with disjunctive queries and it is an extension of Bayesian Inference Networks (Ogilvie and Callan 2001). In this proposition, all documents as merged in one big document. By this merging, the score of a collection is high if the occurrences of one term or more, but not necessarily all terms of the query, are high; despite that users are interested by exact responses. This means that CORI gives more importance to quantity than to quality of responses.

Similarity between a query and a collection is the sum (disjunction) of probabilities of query terms which belong to

$$CORI\ (q,c) = \frac{\sum_{t \in q\ \&\ c}(d_b + (1-d_b).\ T_{c,t}\ .\ I_{c.t})}{|Q|} \qquad (1)$$

where:

$T_{c,t}$ is the weight of term *t* in the collection. $T_{c,t}$ is calculated in the following equation:

$$T_{c,t} = d_t + (1-d_t)\ .\frac{f_{c,t}}{f_{c,t} + 50 + 150 \cdot F_c / \bar{F}_c} \qquad (2)$$

$I_{c,t}$, the inverse collection frequency, is given by the next equation:

$$I_{c,t} = \frac{log\ ((N\_0.5)/f_t)}{log\ (N+1.0)} \qquad (3)$$

Table1 gives the significations of the symbols used in equations (1), (2) and (3).

| Symbol | Signification |
|---|---|
| $\|Q\|$ | The number of terms in the query $Q$ without redundancy |
| $d_b$ | For default belief, is a heuristic value, generally it is equal to 0.4 |
| $d_t$ | For default term frequency, is a heuristic value, generally set to 0.4. In (Callan 2000, French et all 1999) was set to 0. So, there is no agreement about the final equation of CORI and $d_t = 0.4$ is not necessarily correct. |
| $f_{c,t}$ | The number of documents of *c* where *t* occurs |
| $F_c$ | The number of terms of the collection $c$<br>The average of all $F_c$ of all collections |
| $I_{c,t}$ | The inverse of the collection frequency of $c$ where $t$ occurs |
| $f_t$ | The number of collections which contains the term $t$; |

Table 1. Symbols significations in CORI

It appears from (2) that $T_{c,t}$ depends largely on $d_t$ which is a constant obtained by analyzing the dataset on which the computation will be done. As $d_t$ is the minimum term frequency and as it is a fixed value for the system, so all collections are considered the same towards the query according to terms; in other words, the number of occurrences of terms do not signify anything. By this logic, we cannot conclude that a collection $c_1$ is more relevant than a collection $c_2$ to $Q$ even if the terms of the query appear more times in $c_1$ than in $c_2$.

Also from (2), $T_{c,t}$ depends on $f_{c,t}$ and $F_c$. The score of a collection is correlated to $f_{c,t}$, the more this term is large the more the score is high. It results that for two collections $c_1$ and $c_2$, if $f_{c1,t} >> f_{c2,t}$ then $T_{c1,t} > Tc_{2,t}$ despite if the frequency of the term *t* in $c_2$ is greater than its frequency in $c_1$. This is not suitable for works which deal with users profiles. We think also that this is not much real as users are interested by collections which contain a maximum of occurrences of queries terms. Thus, we conclude that CORI gives more importance to quantity than to quality.

$T_{c,t}$ is also affected by $F_c$ the size of the collection. The score CORI(q,c) $\frac{1}{F_c}$. So, big collections are penalized. Callan himself recognizes that in (Si and Callan 2003 a). In (Si et al 2003 b) and (Si and Callan 2003), it is shown that when the size of documents varies, CORI does not give good results. These works give methods for estimating the relevant documents but no discussion was given about the choice of parameters. We think that methods based-heuristic are all confronted to the problem of parameterization. Note that the rest of the equation (1) is invariant at the computation moment; so the final score depends on (2). (D'Souza 2004) has shown that for every dataset $d_t$ and $d_b$ must be changed. These changes pose an important problem of relevance to CORI. The factor $I_{c,t}$ has no effect on the calculus as it is the same value of all collections for the same query.

### 3.2. The Collection selection CSmethod

The principal aim of this method is to return responses promptly. (Abbaci et al 2002) propose to reach only to the first *n_doc* documents of every collection. By this way, they avoid to analyze the entire collections contents. The

proposed formula is largely inspired from (Lawrence et al 1998). For queries with more than two terms, CS gives importance only to the two first terms as these ones are considered the most important in the query (Clarke et al 1995). This restriction has an important impact on the final result, as we will show it after:

In CS, the score of a collection $c$ is computed from the scores of the relevant documents of it. The score of a document *d* from the *n_doc* ones is computed as follows:

$$score\,(d,q) = (c1, nb_d) + (c2.\ ind\_dis\ (d,q)) + \frac{nb\_occ}{c3} \quad (4)$$

where $nb_d$ is number of query terms contained in *d*. *nb_occ* the occurrences numbers in *d*. *ind_dis* it is a distance indicator between two terms of the query in *d*. $c_1$, $c_2$ and $c_3$ are positive constants; precisely, $c_1$ = 100, $c_2$=1000 and $c_3$=1000.

Finally, the score of a collection (server) $S_i$ is the sum of its documents scores, as shown in equation (5):

$$S_i = \sum_{x,y\ d \in n\_doc_i} score\,(d,q) \quad (5)$$

CS returns rapidly responses with a minimum amount of messages, but it suffers from some problems. The first one is in the manner of documents selection. The fact of limiting the responses number to *n_doc*, an important number of more relevant documents will be ignored, especially, when this number is greater than *n_doc*. The loss is more important when the ignored documents are rarely found in another collection.

The second problem resides in the calculus of the distance between terms. For queries with three terms or more, authors choose to compute the distance between the two first terms only. Intuitively, for this type of queries, false positive and false negative numbers of documents will grow. For example, suppose the query $Q$ = < *chemical, composition, water*>. According to CS, the distance which will be considered is between the terms "*chemical*" and "composition". Table 2 gives three relevant responses where only one of them is the most relevant. So, if score of this response is less than the other responses scores and if *n_doc* = 2 then it will be discarded.

| Search Results |
|---|
| Chemical Composition Mercury |
| Chemical Composition air |
| Chemical Composition Water |

Table 2. An Example Result for the query$Q$ in CS

## 4. Collection Selection-Based Relevance Degree CSRD

Before presenting our function of collection selection, we give a set of definitions about relevance and relevance degree of collections and document. For our knowledge, it is the first time that such definitions are given. Preliminary work was published as a poster in (Mechach et al 2014) and a second has recently appeared in (Mechach et al 2015).

### 4.1 Definitions of the Relevance and the Relevance Degree

We have been embarrassed about the choice of extra-collection parameters and constants. Methods using such parameters are not stable; they need to be calibrated from time to time according to collections (D'Souza et al 2004). More than this, in certain situations we have difficulties to selecting a response. Let us see the next example:

As an introductive example, suppose a query Q with one term $t_1$. We consider the notation given in section3:

Documents number $f_{c1,t1}$ = 5 in the collections $c_1$;
Occurrences number $F_{c1t1}$=12; the occurrences of $t_1$ in $c_1$;
Documents number $f_{c2t1}$=17 in the collections $c_2$;
Occurrences number $F_{c2t1}$=17; the occurrences of $t_1$ in $c_2$;

In this situation, it results: in one hand, that the collection $c_2$ contains *17* documents where the term $t_1$ appears only one time per document, in the other hand, the concentration of this term in $c_1$ is more important. This reveals that there are some documents where $t_1$ appears many times. In point of view of users, $t_1$ is more relevant than $c_2$. It is sure that CORI returns $c_2$, as its documents number is greater than $c_1$'one. We don't agree with this method as, in considering semantic point of view of the search, information amount in $c_1$ is greater than its homolog in $c_2$.

Our selection function privileges the collections which contain the most relevant documents to a query $Q$. A collection has a best rank if the degrees of its relevant documents are very high. In order to compute the collections ranks, we give the next definitions:

**Definition1: Relevance of a Document:**
A document $d$ is said to be relevant for a query $Q$ if it shares some terms with it. It is calculated by equation (6).

$$d \cap Q \neq \varnothing \quad (6)$$

**Definition2: Total Relevance of a Document:** A document $d$ is said to be totally relevant for a query $Q$ if it contains all terms of $Q$. It is calculated by equation (7).

$$d \cap Q = Q \quad (7)$$

**Definition3: Document Relevance Degree:** The relevance degree of the a document *d*, noted $d^{\circ}(d)$, for a query $Q$ is the number of terms shared between $d$ and $Q$. It is calculated by the next equation

$$d^{\circ}(d) = |d \cap Q| \quad (8)$$

**Definition4: total document relevance degree:** The relevance document degree $d^{\circ}(d)$ is total, and noted $td^{\circ}(d)$, if it verifies the equation (8)

$$td^{o}(d) = |Q| \quad (9)$$

**Definition 5: Relevance of a Collection:** A collection $c$ is relevant to a query $Q$ if it contains one, or more, relevant document. We compute it by equation (10)

$$d^{o}(c) = |c \cap Q| \quad (10)$$

Let $d_i$ and $d_j$ two relevant documents to the query $Q$. $d_i$ is said to be more relevant to $Q$ than $d_j$ if and only if $d°(d_i) > d°(d_j)$. By the same way, suppose two relevant collections $c_i$ and $c_j$. $c_i$ is more relevant to $Q$ than $c_j$ if $d°(c_i) > d°(c_j)$.

### 4.2 Collection Selection Based-Relevance Degree CSRD

As we have said before, we compute, on the fly, all possible hidden clusters in the selected collections. As such, we give importance to the anatomy of the collections. The function collection selection CSRD is based on the calculus of the relevance degrees of documents and collections. The more a relevance degree is high the more its correspondent document or collection has high chance to be highly ranked. For this reason, we call it CSRD for Collection SelectionBased-Relevance Degree. We present it in the rest of this section.

On receiving the query $Q$, the server $S_i$, which possesses the collection $c_i$, consults its local index term/documents in order to deduce the relevant documents and executes the following steps. The objective is to return a response to the broker:

1- Locating all relevant documents $d_j$ such $d_j \cap Q \neq \varnothing$ (according to definition (6))

2- Extracting all total relevant documents $d_k$ such $d_k \cap Q = Q$ (according to definition (7))

*3-* $S_i$ calculates the score of its collection according to the following equation:

$$f(Q, c_i) = \left( \frac{1}{Nd_{j,i}} + Nd_{k,i} \right) * \sum_{t=1}^{|q|} TF_{t,i} \quad (12)$$

where: $Nd_{j,i}$: is the number of relevant documents of type $d_j$ in step 1; $Nd_{k,i}$: is the number of documents of type $d_k$ obtained in step 2; $TF_{t,i}$: the term frequency of $t$ in $c_i$;

When $S_i$ defines completely the score of $c_i$, it sends it to the broker. This one merges all collections in one list after what it sorts this list in a descendant order. In this manner, the list will contain, in the first ranks, the responses to the conjunctive responses, where it is necessary to respond exactly to queries. The rest of the list is composed by all possible combinations of the query terms according to their concentration in the documents. It is clear in equation (12) that no extra-collections parameters are defined and the sizes of collections have no effect, we give importance to the appearances of terms. CORI responds to disjunctive queries without any semantic.

## 5. Experimentations

### 5.1 The Programming Environment

The experimentations were conducted on a machine with Intel® Core™2 Duo CPU T6570@ 2.1Ghz and with 3GO of AM. We have implemented the three methods in order to compare them. The programs were written in Java in Netbeans 7.0. We have used the Reuters RCV1(Lewis et al 2004).

### 5.2 The simulated system

We have implemented a distributed system as described in (Callan 2000). A central manager or broker manages a set of other servers (peers). Like in (Doulkredis et al 2008) the files of the dataset were randomly distributed on peers. Every set of documents constitutes the collection $c_i$ of the server $S_i$. The broker detains an inverted index Term/Server and uses it for selecting the most relevant servers to a query Q. This manner of selection avoids the broker to send the query to the inappropriate servers. Figure1 presents the selection and evaluation processes of a query. Our collection selection scheme is different from (Hawking et al 2000). This one proposes to send a probe query to all servers. After, a packet of query information is returned. The processes must cooperate in order to merge results.

Every server detains a collection and indexes it locally. A local index is a set of dynamic lists Term/Documents which determine for every term the list of documents where it appears with its frequencies. This inverted list permits to locate the relevant documents rapidly.

### 5.3 Evaluation Methodology

At the first time, we compare the three methods on their accordance to the relevance definition given in section 4.1. In this stage of comparison, we execute a query composed of three terms. This example serves as a prolog for explaining the comparison methodology of the second time. So, we test their relevance and we compare them according to the precision and the F-measure metrics.At the second time of experimentations, we compare the three methods on the precision, the recall and the F-measure metrics. Here, we quantify the executions and we comment the results.Detailed definitions and formulas of these measurements can be found in (Manning et al 2008) and (Büttcher et al 2010).

| Collection | $d°$ | $td°$ |
|---|---|---|
| 153 | 2 | 3 |
| 47 | 1 | 2 |
| 141 | 3 | 2 |
| 40 | 7 | 1 |
| 170 | 6 | 1 |
| 6 | 6 | 1 |

Table 3.The best six results for the query Q = <*bahia, cocoa, july*>

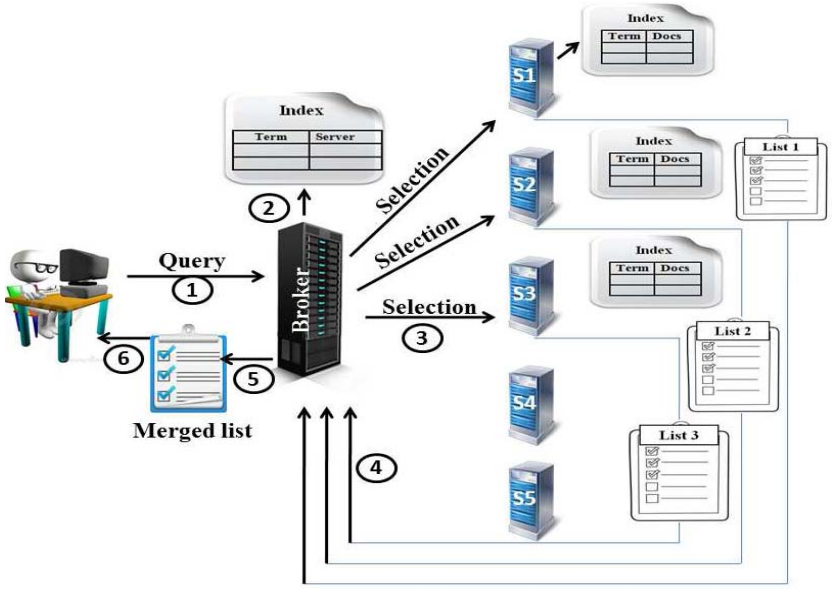


Figure 1. The evaluation process of a query in our proposition

| CORI | | CS | | CSRD | |
|---|---|---|---|---|---|
| **Collection** | **Score** | **Collection** | **Score** | **Collection** | **Score** |
| 6 | 0.12439708 | 104 | 0.6275 | 153 | 202.91 |
| 153 | 0.12436888 | 165 | 0.5776 | 47 | 126.21 |
| 40 | 0.12430853 | 3 | 0.5715 | 141 | 85.55 |
| 47 | 0.12430804 | 114 | 0.3895 | 155 | 83.53 |
| 189 | 0.12430578 | 116 | 0.3262 | 40 | 76.17 |
| 82 | 0.12430570 | 130 | 0.2923 | 102 | 74.82 |

Table 4.The six best collections and their scores returned by the three methods for the query Q

### 5.4 Comparison on one example

In this experience, we send the query $Q = <bahia, cocoa, july>$ into a centralized system in order to collect the set of pertinent collections. We collected also the degrees $d°$ and $td°$ values (see section 4.1). We refer to this as the real existence. Table 3 gives a fragment of 6 results. The first column presents the pertinent collections in the system. The second column gives the degrees relevance while the total degrees are given in the third column.

Table 3 shows, clearly, that the collections with high total degrees are given in the best ranks (the first three lines). In other terms, the importance is given to the conjunctive responses. The rest of responses are ranked according to relevance degrees. We assume that this is a logical order.

At the second time, we send the same query $Q$ through a simulated system composed of 200 peers. Table 4 gives the results obtained from the three methods. We see that CORI gave collection 6 as the best response. On analyzing this collection, we found that it has not a highnumber of terms $F_c$ in comparison of the rest of collections also the number $f_{c,t}$ of this collection is less than the other $f_{c,t}$ s of the other collections (see equation (2)). For these reasons, it is well ranked. Globally, CORI has returned good results. We must note here that we have chosen the best result that we have found for this method. Unluckily, CS has not returned good results in comparison of the other methods. This method depends largely on *n_doc*, which delimits the number of documents to return, and on the distance between the first two terms. So its responses cannot be so exhaustive. Table 3 shows that CSRD is quite similar to the real existent. CSRD has returned the best responses in the best ranks. Collection 47 is ranked better than collection 141 because we have divide by $Nd_{j,i}$ in equation (11). But despite this division, we privilege the conjunctive queries in our approach, and this is quite the same as the introductive example given in section 4.1.

### 5.5 Comparison on the Precision on $Q$

We have computed the precision of the search in the centralized system. Table 5 gives the best 10 values of precision. Figure 2 depicts the precision values of the collections subject to comparison.

| Collection | Precision |
|---|---|
| 153 | 0.00454 |
| 47 | 0.00304 |
| 141 | 0.00285 |
| 40 | 0.00164 |
| 156 | 0.00160 |
| 43 | 0.00159 |
| 53 | 0.00157 |
| 48 | 0.00157 |
| 82 | 0.00156 |
| 197 | 0.00154 |

Table 5. Thereal Top-10 collections and their precision values for the query <*bahia, cocoa, july*>

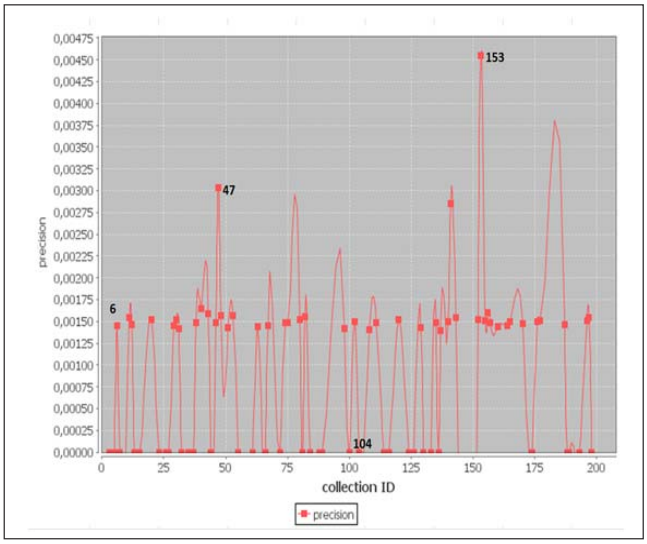


Figure 2. Precision curve of the query Q=<*bahia, cocoa, july*>

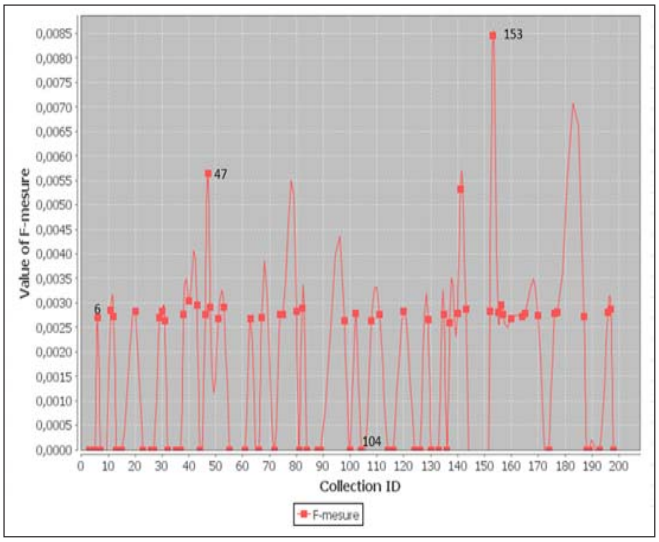


Figure 3. F-measure curve of the query Q=<*bahia, cocoa, july*>

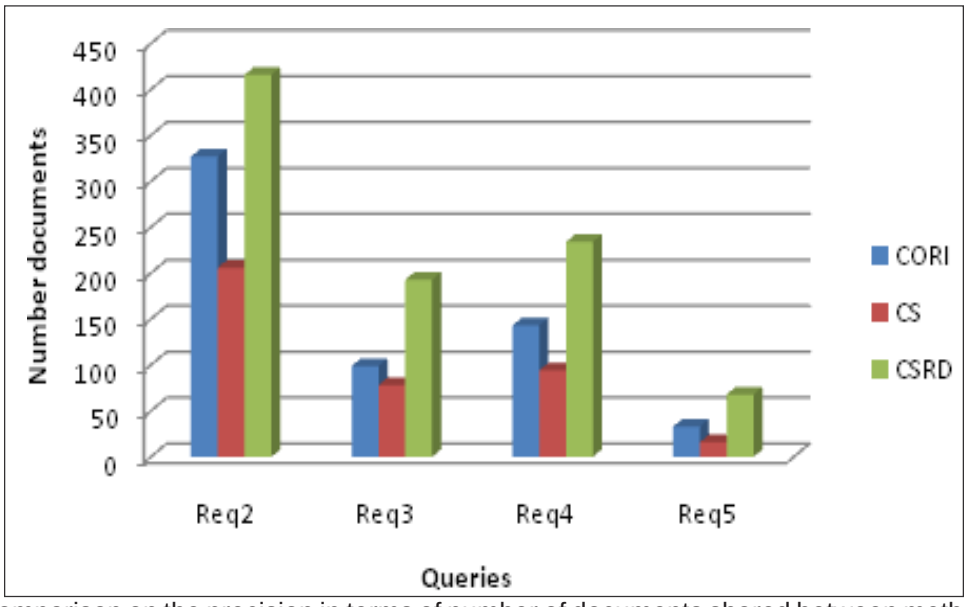


Figure 4. Comparison on the precision in terms of number of documents shared between methods and the centralized system

Figure2 shows that CSRD's top-2 has the best values precision(collections 153, 74), while CORI's Top-1 precision is not globally high. At the third place, we found that the CS has not returned collection with high precision.

### 5.6 Comparison on the F-measure on Q

We have also computed the F-measure for this query. We omit here the recall, as F-measure contains it with the precision, but we will consider it for the quantification of experimentations in the next section. Figure3 shows clearly that CSRD has returned the best responses with the best f-measure. In the second place, CORI has also returned good results but CS is not at the same level of responses quality as the precedent methods.

This study is composed of two parts. In the first one, we have studied what exist really in the system. Thus, we have taken the whole collection like in a centralized server and we have executed 4*1000 queries. The lengths of these queries varied from two to five terms. So, Req2 designates that the query Req is of two terms. We have collected for every query Req the set of relevant documents in the system and their precisions. The relevant documents share, at least, one term with Req. These lists are sorted in a descendant order on precisions. So these sets are really the best responses for the 4000 queries.

In the second part of this experimentation, we have executed these 4*1000 queries in our distributed system. We have studied especially the Top-5 for every method, as the number of responses was important. We have calculated the intersection between the Top-5 of every method, for every type of queries, with the existent in the centralized system.

The result of this experience is depicted in figure 4. We can see that CS has returned the lowest number of responses in Top-5. It has located less number of documents from the real relevant documents. The curve of CS_3 (in yellow color) cannot appear as its values were practically equal CS_2' ones. In the second place, we found CORI as it shares more documents with the real relevant documents. This experimentation gives confirmation that CSRD has returned more relevant documents in the best ranks. It is clear that CSRD has returned double of the number of relevant documents returned by CORI.

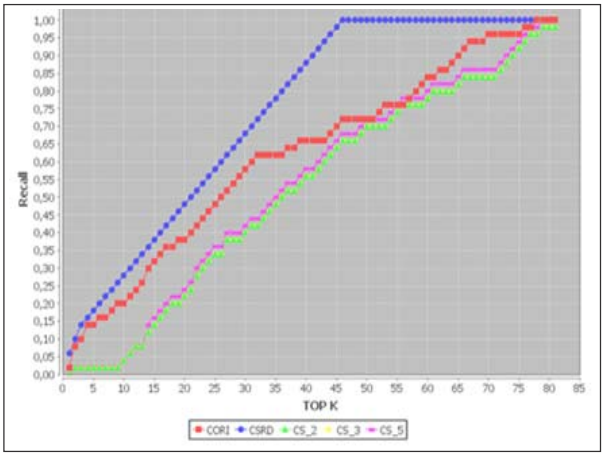


Figure 5. Comparison in term of the recall between the three methods

### 5.8 Analysis of the Results in Term of the Recall

In this experimentation, we compare these three methods according to the recall measure.

Figure 5 depicts the recall of these three methods. We have executed intensive experimentations by varying Top-k. We see on this figure that CSRD presents a high performance; its recall values are better than those performed by CORI and CS. CRSD has reached the maximal value recall=1 early for k=46.

We have executed CS with the values 2, 3, and 5 for *n_doc*. We can see on figure5 that the more *n_doc* is high the more CS presents a best recall. For this method, recall is proportional to *n_doc*. With the values *n_doc=2* and 3, the recall cannot reach 1; in other words, there are some relevant documents and rare which the system cannot achieve. We see also that CS has reached 1 for *n_doc=5*, which is logic; but the CS's designers do not like such values for *n_doc*. This figure shows also that CORI has returned more results than CS as its curve is above the CS's curve; however, it cannot achieve CSRD performance.

### 5.9 Comparison According to the F-measure

In this section, we compare these methods according to the F-measure metric. This metric includes the recall and the precision metrics.

Like in the precedent experimentation, we have executed 4*1000 queries. The queries are composed of 2, 3, 4 and 5 terms. The executions were done in a centralized system in order to collect all relevant and exhaustive responses. For every query, the results were sorted in a descendant order on the F-measure. We have injected and executed these queries in our distributed system. We have collected the intersection between the F-measure of the centralized and the distributed systems for the three methods by considering the Top-10.

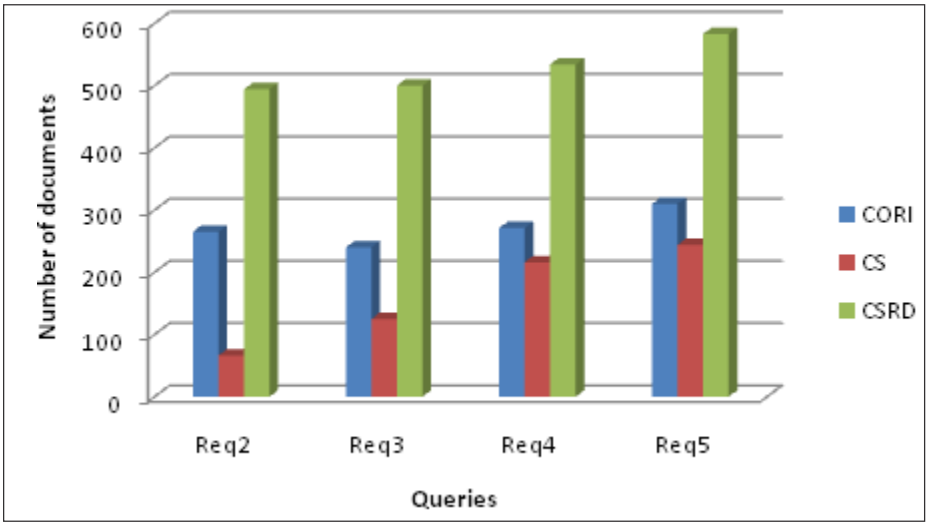


Figure 6.Comparison on the f-measure in terms of best responses returned by methods comparatively to the centralized system

Figure6 shows that CSRD has shared more results with the centralized system, while CORI has produced nearly the half of CSRD's results. CS method has produced fewer results than the other methods. It is penalized by the parameter *n_doc*.

## 6. Conclusion

Collection selection is intensively studied in the information retrieval area. Till these new works are published in order to touch to perfection. The main objective of such study is to respond to users'queries with high probabilities. In this area, one of the popular methods is CORI. This one considers collection as a big document and a document is relevant to a query if it shares terms with it. Ranking relevant collections is done according especially to two (three) parameters: the number of documents in the collection and the number of terms in it. The most the first parameter is high the most the score is high; however, the final score is inversely proportional with the number of terms. The study of the efficiency of this method shows that it is unstable in its ranking process. For different datasets, the heuristically parameters used in the formula score, must be adjusted. We have revealed that this instability is due to the default terms frequency; this default value cannot by the same for all datasets.
We have studied and presented the CS method. The aim of CS is to touch to a little number of servers by reducing the number of documents that will be returned. Surely, this method is rapid and turns a few flows of messages in the system, but it is not so exhaustive; the false and true positive numbers are important.

In order to give a new function for selecting collections in digital libraries, we have given some important definitions to the relevance of documents and collections to queries. For us, a collection is highly ranked if it has an important number of documents similar to query. Our function is free from heuristically parameters. It allows us to surf freely in a digital library. We have given the explanation to the contents of responses by comparing the three methods. The experimentations have shown that our method is efficient and effective according to different metrics.

Actually, we are comparing our method with other methods like (Chernov et al 2006) and on different datasets. Also, we will give a new context in which the CORI method can give better results than its actual version.

## Bibliography

[1] Abbaci, F., Savoy, J., Beigbeder, M. (2002). Methods for collections selection in a distributed system, Congress Documents in mobile information systems, (in French. Méthodes pour la sélection de collections dans un environnement distribué*Congrès Documents dans les systèmes d'information mobiles)*, pp. 227 – 238, Tunisie.

[2] Büttcher, S., Clarke, C. L. A., Cormack, G. V. (2010). Information Retrieval: Implementing and Evaluating Search Engines. MIT Press.P 68.

[3] Callan, J. P., Croft, W. B., Harding, S. M. (1992). The inquery retrieval system, Proceedings of the Third International Conference on Database and Expert Systems Applications, Springer-Verlag, p 78-83.

[4] Callan, J. P., Lu, Z., Croft, W. B. (1995). Searching distributed collections with inference network.*E. A. Fox, P. Ingwersen et R. Fidel, eds, "Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*. ACM, p. 21-28.

[5] Callan, I. P. (2000). Distributed information retrieval. *In W. B. Croft, editor, Advances in Information Retrieval.Kluwer Academic Publishers*. pp. 127-150.

[6] Clarke, C. L. A., Cormack, G. V., Burkowski, F. J. (1995).*Shortest substring rankingMultiText experiments for TREC-4*. TREC'4, NIST Publication p. 295-304.

[7] Chernov, S., Serdyukov, P., Bender, M., Michel, S., Weikum, G., Zimmer. C. (2006). Database Selection and Result Merging in P2P Web Search. *International Workshops, DBISP2P 2005/2006 Trondheim, Norway,*

*August 28-29*, *Seoul, Korea, September 11,* p. 26-37.

[8] Craswell, N., Bailey, P., Hawking, D. (2000). Server selection on the world wide web.*Proc. Fifth ACM Conf. on Digital Libraries*, p. 37–46.

[9] Souza, D. D'., Zobel, J., Thom. J. A. (2004). Is CORI Effective for Collection Selection? An Exploration of Parameters, Queries, and Data.*In Proc of the 9th Australasian Document ComputingSymposium, Melbourne, Australia*.

[10] Doulkeridis, C., Norvag, K., Vazirgiannis, M., (2008). *Peer-to-Peer Similarity Search over Widely Distributed Document Collections. LSDS-IR '08,California, USA*

[11] French, J.C., Powell, A. L., Callan, J., Viles, C. L., Emmitt, T., Prey, K. J., Mou, Y. (1999).Comparing the performance of database selection algorithms.*M. Hearst, F. Gey et R. Tong, eds, Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval, ACM*, p. 238–245.

[12] Hawking, D., Thistlewaite, P., (1999). Methods for Information Server Selection.*ACM Transactions on Information Systems*, 17 (1), p. 40-76.

[13] Holz, F., Witschel, H. F., Heinrich, G., Heyer, G. Teresniak. S. (2007). An Evaluation Framework for Semantic Search in P2P Networks.*the I2CS 2007*.

[14] Jie, L., Callan, J. (2004). Merging retrieval results in hierarchical peer-to-peer networks.*SIGIR'04: Proc of the 27th annual international conference on Research and development in information retrieval, ACM*, p. 472-473.

[15] Jie, L., Callan, J., (2006). User Modeling for Full-Text Federated Searchin Peer-to-Peer Networks. *SIGIR'06, August 6–10, Seattle, Washington, USA*.

[16] Khalaman, S., Kurland, O., (2012). Utilizing Inter-Document Similarities in Federated Search, *SIGIR'12, Portland, Oregon, USA. ACM*, p 1169-1170.

[17] Lawrence, S., Giles. C. L. (1998). Inquirus, the NECI meta search engine,*WWW'7*, p. 95-105.

[18] Manning, C. D., Raghavan, P., Schütze, H. (2008). An Introduction to Information Retrieval. Cambridge University Press. P 155.

[19] Mechach, K., Zekri, L., Adi, M. K. (2014). Une Nouvelle Approche pour la Sélection de Collections dans les Systèmes Distribués. *In Proc of COSI'2014, Bejaia, Algeria*, pp. 25-27.

[20] Mechach, K., Zekri, K., L., Adi. M. K. (2015). Etude de La Pertinence lors de La Sélection de Collections dans les Systèmes Distribués. EGC-2015, Luxembourg, vol. RNTI-E-28, p.489-490

[21] Meng, W., Yu, C., Liu, K. L. (2002): Building efficient and effective metasearch engines.*ACM Computing Surveys* 34 (1), pp. 48–89.

[22] Ogilvie, P., Callan, J. (2001).The Effectiveness of Query Expansion for Distributed Information Retrieval. *ACM-CIKM'2001,* p. 183-190.

[23] Page, L., Brin, S., Motwani, R., Winograd. T., (1992). The pagerank citation ranking: Bringing order to the web. *Technical Report 1999-66, Stanford InfoLab, November*.

[24] Salampasis, M., Giachanou, A. Paltoglou, G. (2013). Multilayer Collection Selection and Search of Topically organized Patents. *In: Proceedings of the Integrating IR technologies for Professional Search Workshop*, *Moscow, Russia, 24-March*, p 48-56,

[25] Si, L., Callan. J., (2003). a. Relevant Document Distribution Estimation Method forResource Selection. In Proc. of the 26th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval.July 28-Aug 1, Toronto, Canada.

**Mechach Kheira** has received her BS Master from department of computer sciences of faculty of exact and applied sciences of university of Oran1, Algeria. Currently, she is preparing her doctorate. She works in the field of information retrieval in large systems.

**Zekri Lougmiri**is assistant professor in the department of computer sciences of the faculty of exact and applied sciences of the university of Oran1, Algeria. He is a permanent member in the laboratory LAPECI. He works in information retrieval in large systems. He works also on peer-to-peer, distributed computing and multi-criteria optimization

**M.K. ABDI** holds a master degree and a Ph.D. degree in computer science from Department of Computer Science at the University of Oran1, Algeria. He is currently, professor for the same Department. His research interests include the application of artificial intelligence techniques to software engineering, software quality, formal specification, Data-Mining and Information Research.