World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:5, 2016

Selection of Relevant Servers in Distributed Information Retrieval System

Benhamouda Sara, Guezouli Larbi

***Abstract*—**Nowadays, the dissemination of information touches the distributed world, where selecting the relevant servers to a user request is an important problem in distributed information retrieval. During the last decade, several research studies on this issue have been launched to find optimal solutions and many approaches of collection selection have been proposed. In this paper, we propose a new collection selection approach that takes into consideration the number of documents in a collection that contains terms of the query and the weights of those terms in these documents. We tested our method and our studies show that this technique can compete with other state-of-the-art algorithms that we choose to test the performance of our approach.

***Keywords*—**Distributed information retrieval, relevance, server selection, collection selection.

I. Introduction

THE growth of the Internet and digital libraries increased the attention to the problem of information retrieval in a distributed environment, where the resources of information are becoming increasingly important, and their contents cannot be explored, and indexed by a centralized information retrieval system.

Distributed information retrieval is an effective solution to these problems, where the task is to find the information needed by a user in an efficient manner. Thus, in the search part of distributed information retrieval we need to select the relevant resources and submit an appropriate query for them [8], [9].

A distributed information retrieval system which conducts research on a set of distributed collections of documents is usually composed of a broker and a set of servers. These servers are all available and known by the broker, and each server has a collection of documents and an information retrieval system that ensures the search in this collection. Upon the receipt of a request, the broker selects a subset of servers among those he knows in an efficient manner, so he can determine the set of servers that are most likely relevant to respond to the request of the user. This transaction, which represents the server selection feature, is called server selection or collection selection [3]-[9].

In this paper, we focus on the problem of collection selection, which is an important task of information retrieval in a distributed environment, where we present our proposed collection selection method. For each entered query, our method ranks collections according to the weights of the query terms in documents of a collection that contain those terms, the number of these documents and the number of query terms which emerged in this collection. If the query terms present a high weight in documents of a collection compared to other collections, the collection should be well classified. By the tests of our approach's performances, the experiments show that our technique can compete with other state-of-the-art methods that we choose for the tests.

Benhamouda Sara is with the University of Batna, Algeria.
Guezouli Larbi is with LASTIC Laboratory of Batna University, Algeria (corresponding author, e-mail: www.larbiguezouli.com).

II. Related Works

Finding the best collection selection strategy is a complex optimization problem and several approaches are proposed in the literature. In this section, we describe the principle of collection selection methods; CORI [2], CVV [5] and vGlOSS [4].

A.CORI

CORI is a collections ranking method, by using a Bayesian inference network to classify these collections [2].

The CORI method uses document frequency (DF) and collection frequency (SF), where each query term is evaluated separately, and the probabilities are calculated using document frequency DF and the inverse collection frequency $I_{si,t}$ [6], [7].

The inverse collection frequency $I_{si,t}$ can be calculated as:

$$I_{s_i,t} = \frac{log\left(\frac{|S|+0.5}{SF_k}\right)}{log(|S|+1)} \quad (1)$$

The document frequency $T_{si,t}$ can be calculated as:

$$T_{s_i,t} = d_t + (1 - d_t) + \frac{log(DF_{i,k}+0.5)}{log\left(DF^{max}_{i,k}+1\right)} \quad (2)$$

The belief is calculated using:

$$p(t_k|s_i) = d_b + (1 - d_b).T_{s_i,t}.I_{s_i,t} \quad (3)$$

The ranking score of the collection for the query q is the sum of all beliefs $p(t_k|s_i)$, where $t_k \in q$:

$$CORI(q, s_i) = P\left(t_1 \dots t_{|q|}\middle|s_i\right) = \frac{\sum_{k=1}^{|q|} p(t_k|s_i)}{|q|} \quad (4)$$

where, S: All known servers by the broker of search (s_1, s_2, ...), |S|: The total number of servers, SF_K is the number of servers where $s_i \in S$ and $DF_{i,k} > 0$., s_i : the search server, q: The query (t_1, t_2, ...), |q|: The number of query terms, t_k : the k^{th} term of query q, $DF_{i,k}$: The number of documents containing the term t_k

in the server s_i, DF^{max}_i: The maximum *DF* of all terms in the server s_i, d_b: The default belief is fixed at 0.4, d_t: The default frequency of the term is fixed at 0,4, $T_{si,t}$: The weight of the term t in the collection s_i, $I_{si,t}$: The inverse frequency of collection [5].

B.CVV

CVV "Cue Validity Variance" is a server selection method based on the CVV of terms of a given query. The objective of the CVV method is to identify collections with a high concentration of query terms, which weighs the terms according to their power of discrimination between collections. [1]-[9]. Terms that can discriminate servers have a higher CVV value and thus they contribute to give a high score to the server s_i [5].

The technique begins with the calculation of the cue validity of each term t_j in each collection s_i, or CV (t_j, s_i) or $CV_{i,j}$ [6]:

$$CV_{i,j} = \frac{\frac{DF_{i,j}}{SS_i}}{\frac{DF_{i,j}}{SS_i} + \frac{\sum_{k\neq i}^{[S]} DF_{k,j}}{\sum_{k\neq i}^{[S]} SS_k}} \quad (5)$$

where $CV_{i,j}$ is the cue validity of the term t_j in the collection s_i. The CVV is the variance of CV across all collections.

$$CVV_j = \frac{\sum_{i=1}^{|S|}(CV_{i,j} - \overline{CV}_{i,j})}{|S|} \quad (6)$$

where; $\overline{CV_{i,j}}$ is the average of $CV_{i,j}$ on all servers.

The score of a server is:

$$SS_i(q) = \sum_{j=1}^{|q|} CVV\ j \cdot DF_{i,j} \quad (7)$$

where; S: All known servers by the broker of search (s_1, s_2, ...), |S|: The total number of servers, SF_K is the number of servers where $s_i \in S$ and $DF_{i,k} > 0$, s_i: the search server, q: The query (t_1, t_2, ...), |q|: The number of query terms, t_k: the k^{th} term of query q, $DF_{i,k}$: The number of documents containing the term t_k in the server s_i, SS_i: The number of documents in the server s_i [5].

C.vGlOSS

vGlOSS is a method of servers ranking where each server uses an information retrieval system based on the vector model [6]. vGlOSS assumes that users are not interested in all documents containing query terms but by the documents which have the similarity with the query is greater to a threshold [6].

vGlOSS search to recover more information about the content of servers.

One of the proposed solutions is to use matrices:

- $F = (f_{ij})$: f_{ij} is the number of documents in the collection containing the term t_j.
- $W = (w_{ij})$: w_{ij} is the sum of the weights of term t_j over all documents in the collection s_i.

The weight of a term t_j is distributed evenly over all the documents that contain it. This means that t_j has the weight:

$$\frac{w_{i,j}}{f_{i,j}} \quad (8)$$

In all documents of the server s_i that contain the term t_j. With this assumption, two scenarios are available: vGlOSS scenario with high correlation and vGlOSS disjoint scenario.

1.vGlOSS Scenario with High Correlation

vGlOSS suppose that if two words occur together in a user's request, these terms appear in the collection of documents with the highest possible correlation.

Calculating the estimation of similarity between the query q and each server according to a certain threshold is given by:

$$Estimate\ (l,q,s) = \sum_{j=1}^{p}(f_{i,j} - f_{i,j-1}) \times sim_j \quad (9)$$

where:

$$sim_j = \sum_{k=j}^{|q|} q_k \times \frac{w_{i,k}}{f_{ik}} \quad (10)$$

l: The threshold, q_k: The frequency of term t_k in the query q, p: The order of the term t_p in the query q defined as:

$$sim_p = \sum_{k=p}^{|q|} q_k \times \frac{w_{i,k}}{f_{ik}} > l \quad (11)$$

$$sim_{p+1} = \sum_{k=p+1}^{|q|} q_k \times \frac{w_{i,k}}{f_{ik}} \leq l \quad (12)$$

i.e. $sim_p > l$ et $sim_{p+1} \leq l$ (the f_{ij} must be sorted $0<f_{i0}<f_{i1}<\ldots$); where |q|: The number of query terms.

2. vGlOSS Scenario Disjoint

If two terms t_1 and t_2 appear together in a query q, these terms should not appear together in a document of the selected server.

The estimate of similarity between a server and the query is:

$$Estimate\ (l,q,s) = \sum_{k=1..|q|,(f_{ik}>0)\wedge(q_k\times\frac{wik}{fik}>l)}\ q_k \times w_{ik} \quad (13)$$

where; l: The threshold, q_k: The frequency of term t_k in the query q, |q|: Number of query terms, f_{ik}: The number of documents in the collection s_i that contain the term t_k, w_{ik}: The sum of the weight of term t_k in all documents of the collection s_i [4].

III.Proposed Approach

In this section, we are interested in presenting our proposed approach.

A.Research Questions

We state the assumptions where our approach is based on three hypotheses to clarify the problem:

Hypothesis 1: For a query term, the weight distribution on documents containing that term and the number of those documents in the server s_i, are important.

Hypothesis 2: Take into account the number of query terms that appears in the server s_i, we assume that the query terms appear in a server, are in the same document.

Hypothesis 3: If the query terms occur with a high weight in documents of a collection compared to other

collections, this collection should be well classified.

B. Proposed Approach

1.Calculate the Weight Distribution

The weight distribution of the query term t_k on documents of the server s_i containing this term is calculated using the following formula:

$$dist\ (t_k, s_i) = \frac{w_{i,k}}{DF_{i,k}} \quad (14)$$

2.Calculate the Number of Query Terms

The number of query terms that appears in the server s_i is calculated using the formula:

For all k in {1, .., |q|}, if ($w_{i,k}$!=0)

nb_terme_requete(s_i) = nb_terme_requete(s_i)+1.

where |q| is the size of the query.

3. Calculate the Number of Documents Containing the Query Terms in the Server s_i

$$DF_i = \sum_{k=1}^{|q|} DF_{i,k} \quad (15)$$

4. Calculation of the Similarity between the Query q and the Collection s_i

Calculate the score of a server to a posed query by using the formula:

$$S_{ci}(q, s_j) = \frac{nb\_terme\_request(si)}{|q|} \times DF_i \times \sum_{k=1}^{|q|} dist(t_k, s_i) \quad (16)$$

where; dist(t_k,s_i): The weight distribution of the query term t_k on the documents containing that term in the server s_i, nb_terme_request(s_i): the number of query terms that appears in the server s_i, DF_i : The number of documents containing the query terms in the server s_i.

IV. Experimental Studies

After the description of our proposed approach to the problem of selecting relevant servers in a distributed environment, we present the results of experiments that conduct to test the performances of our proposed approach. These tests are based on comparing the performances of our proposed approach with the performances of collection selection methods CORI, vGlOSS scenario with high correlation, vGlOSS scenario disjoint, and CVV, that we have describe their principles.

To make this comparison we study for each method and our proposed approach: the evolution of the selection time relative to the number of available servers. We also study the returned selection results to the posed queries, by the Px and Rx measures which are described below.

A.Our Test Collection

Concerning the test corpora of different servers, to conduct our experiments we used a corpus for each server. A server is represented by the index corpus that is generated by an information retrieval system. Documents are textual documents in the French language.

B.The Queries of Tests

Query 1: "Object oriented programming ".
Query 2: "server's selection in a distributed information system".
Query 3: "Flowers and plants in the environment".

C.Evaluation

To evaluate the performances of our selection approach, we choose to study the change of selection time relative to the number of available servers. We also use the evaluation Px and Rx metrics which are defined as:

- Px determines the proportion of the relevant servers compared to the selected servers (x):

$$P_x = \frac{|NbServPert_x|}{x} \quad (17)$$

such as |NbServPertx|: The number of relevant servers from selected x servers.

- Rx: determines the proportion of relevant documents associated to the selected servers when x servers are selected:

$$R_x = \frac{\sum_{i=1}^{p} r_i}{Pert} \quad (18)$$

such as r_i: The number of relevant documents in the server s_i, Pert: The number of relevant documents in the set of servers in the system.

D.Evaluation and Comparison

1.Evaluation and Comparison by Selection time

The query used for the test of selection time is:

Query 2: "Selection of servers in a distributed information retrieval system".

- By examining Fig. 1, we note that:

The results show that the selection time of our proposed approach increases linearly with the increasing of the size of the test set, and we note that it is the shortest time compared to the four other methods, because of the simplicity of the calculations of our proposed approach.

We conclude that the selection time is relatively related to the number of available servers for our proposed approach.

For a corpus size of 100 servers (index) we obtained the following results:

- CORI: 363 milliseconds.
- CVV: 3629 milliseconds.
- VGlOSS scenario with high correlation: 14 milliseconds.
- Disjoint vGlOSS scenario: 4 milliseconds.
- Proposed approach: 3 milliseconds.

- By examining Fig. 2, we note that:

The results show that our selection approach proposed presents Px is equal to 100% for a selected number of servers equal to 1. The same remark for the four selection methods, and gradually the measure Px will decrease because of the

proportion of relevant servers to selected servers. We note that the performance of our approach is similar to that of CORI and our approach gives better results compared to the results of the three other selection approaches: vGlOSS scenario with large, vGlOSS scenario disjoint, and CVV, i.e. the proportion of one of the relevant selected servers by our approach is the largest compared to the other three approaches.

- By examining Fig. 3, we note that:

The results show that our proposed approach Rx increases to 100%, and the proportion of relevant documents (belonging to the selected servers) increases with the increasing of the number of selected relevant servers. The same can be said for the four other selection methods, and we also note that our approach gives better results compared to the results of selection approaches: vGlOSS scenario with high correlation and vGlOSS disjoint, i.e. the number of relevant documents belonging to selected servers by our approach among existing relevant servers is high compared with these two approaches.

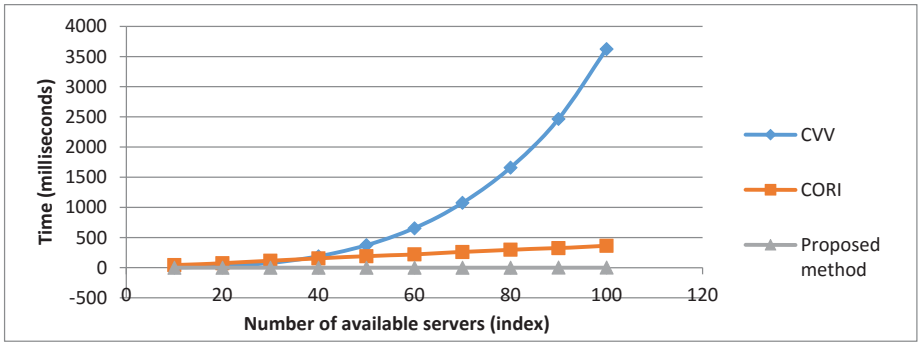

Fig. 1 Comparison between the proposed approach with the collection selection approaches (CORI, CVV, vGlOSS scenario and high correlation, vGlOSS disjoint scenario) using the evolution of time selection related to the number of available servers

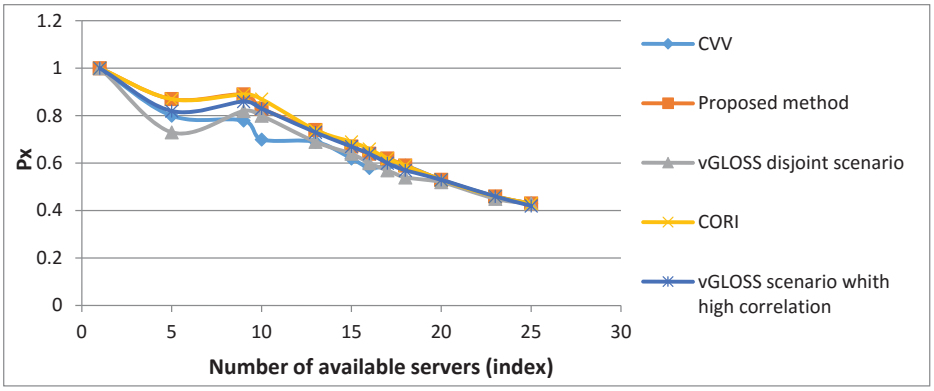

Fig. 2 Comparison between the proposed selection approach and the selection approaches (CORI, CVV, vGlOSS scenario with high correlation vGlOSS disjoint scenario) using the Px metric for the overall results returned for the three queries

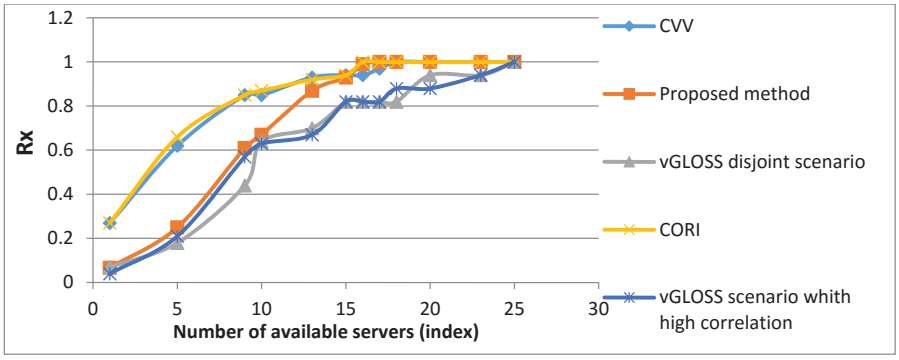

Fig. 3 Comparison between the proposed selection approach and the selection approaches (CORI, CVV, vGlOSS scenario with high correlation, vGlOSS disjoint scenario) using the Rx metric for the overall results returned for the three queries

V. Conclusion

Our approach takes into account the weight distribution of the query terms in the documents containing these terms and the number of documents containing query terms in a server, i.e. the query terms occurring with a high weight in documents of a collection compared to other collections, the collection should be well classified. In this paper we have reported the following conclusions based on our experiments:

- Concerning the proportion of the relevant servers compared to selected servers, the performance of our approach and CORI are similar, and it gives better results compared to the results of the CVV vGlOSS scenario with high correlation and vGlOSS disjoint scenario approaches.
- In terms of proportion of relevant documents belonging to the selected servers compared to existing relevant documents, our approach also gives better results compared to the results of scenario approaches: vGlOSS with high correlation and vGlOSS disjointed.

References

[1] Allison L. Powell, and James C. French, "Comparing the Performance of Collection Selection Algorithms", In ACM Transactions on Information Systems (TOIS), Vol.21, No.4, 2003, pp. 412–456.

[2] Daryl D'Souza, Justin Zobel, and James A, "Is CORI Effective for Collection Selection an Exploration of Parameters, Queries, and Data", In Proceedings of the Australian Document Computing Symposium, Melbourne, Australia, December 2004, pp.41-46.

[3] Faïza Abbaci,"Méthodes de sélection de collections dans un environnement de recherche d'informations distribuée", Thesis, Neuchâtel University, 2003.

[4] Luis Gravano, Héctor Garcia-Molina and Anthony Tomasic, "GlOSS: Text-Source Discovery over the Internet", In Journal ACM Transactions on Database Systems (TODS), vol. 24, no. 2, Jun 1999, pp. 229-264.

[5] Nicholas Eric Craswell, "Methods for Distributed Information Retrieval", Thesis, Australian National University, 2000.

[6] Paul Thomas, and David Hawking, "Server selection methods in personal metasearch: a comparative empirical study", In Information Retrieval Jounal, Vol. 12, Issue 5, 2009, pp. 581-604.

[7] Sander Bockting, "Collection Selection for Distributed Web Search Using Highly Discriminative Keys, Query-driven Indexing and ColRank", Thesis, University of Twente Enshed-The Netherlands, 2009.

[8] Umberto Straccia, and Raphaiel Troncy, "Towards Distributed Information Retrieval in the Semantic Web: Query Reformulation Using the oMAP Framework", The 3rd European conference on the Semantic Web: Research and Applications Lecture Notes in Computer Science, 2006, pp. 378-392.

[9] Fabio Crestani, Ilya Markov, "Distributed Information Retrieval and Applications", In the 35th European Conference on IR Research (ECIR 2013), Moscow, Russia, 2013, pp. 865-868.

Sara Benhamouda received a bachelor's degree in sciences from the high school of Batna, Algeria, in 2009 and a master's degree in computer science from the University of Batna, Algeria, in 2014.

Larbi Guezouli (b. 1974) received a bachelor's degree in mathematical techniques from the Technical High School of Batna, Algeria, in 1992, and an engineering degree in computer science from the University of Batna, Algeria, in 1997 and Ph.D. degree in computer science from the University of Paris 7 (Denis Diderot), France, in 2007. He is currently also a Deputy Director of the computer science department at the University of Batna in Algeria. His major interests are information retrieval, data mining, and cross language. He is the author of more than 9 conference proceedings papers, 4 journal papers, and a patent in his research areas.