The Collaborative relevance in the distributed information retrieval

Adil ENAANAI
TIES team, ENSIAS
Mohamed V University
Rabat, Morocco
enaanai@gmail.com

Aziz SDIGUI DOUKKALI
TIES team, ENSIAS
Mohamed V University
Rabat, Morocco
doukkali@ensias.ma

Ichrak SAIF
TIES team, ENSIAS
Mohamed V University
Rabat, Morocco
s_ichrak@gmail.com

Hicham MOUTACHAOUIK
I2S2E Research Lab., SISM Team
Department of industry, ENSAM, University
Hassan II
Casablanca, Morocco
gotohicham@gmail.com

Mustapha HAIN
I2S2E Research Lab., SISM Team
Department of industry, ENSAM, University
Hassan II
Casablanca, Morocco
infohain@yahoo.fr

***Abstract*—Relevance is one of the most interesting topics in the information retrieval domain. In this paper, we introduce another method of relevance calculation. We propose to use the implicit opinion of users to calculate relevance. The Implicit judgment of users is injected to the documents by calculating different kinds of weighting. These latter touch several criteria like as user's weight in the query's words, user's profile, user's interest, document's content and the document popularity. In this method, each user is an active element of the system, he searches documents and he makes treatments to provide relevant information to other users in the Network. This is similar as the peer-to-peersystems; unlikethat, an element (user) have tomanage automatically his data by creating a short view model of his most visited documents, and calculates his relative relevance about each one. The relative relevance is variable according each user, so the final relevance is calculated by the averaging of the elementary relevance of all users. Hence, the name of collaborativerelevance.**

Keywords: Relevance; Collaborative; information retrieval.

I. INTRODUCTION

In the big data context, documents have several formats, languages, and kinds of content. In order to exploit existing information, it is necessary to use the Electronic document management (EDM). The existing tools of the EDM provide us some services like as classification, filtering and search[1]. The last is becoming one of the most important services in the web. Simply know that Google now processes over 40,000 search queries every second on average, which translates to over 3.5 billion searches per day and 1.2 trillion searches per year worldwide (*by Google search statistics service*).

Each information retrieval system uses its own method to rank results. So, we can privilege some technics to sort retrieved documents which are based on the in-page or out-page criteria. The combination of these criteria is becoming unavoidable to have an efficient relevance function[2]. In fact, the objective of all Information Retrieval Systems (IRS) is to decrease silence and noise. These two factors mean respectively the absence of relevant documents in the result page, and the presence of the irrelevant documents in the result page[3].

To have an efficient result, user must present his request exactly. The request must be clear, unambiguous and accurate. Each request and document have some intersection points that are needed to calculate similarity function. This last is based on the calculation of distance between document and request with consideration of the user profile[4].

In this paper, we introduce a new model of relevance calculation by defining three levels of relevance. Firstly, we will begin by a global stat of the art,in which we will present some new works in the domain, and review their results. Then, we will compare the different models relevancecalculation and synthesize them. Finally, we will introduce our model and give formalization for it.

II. Problem

Relevance is one of the most complicated problems in the information retrieval systems (IRS). Giving the relevant documents is not a simple task; there are several criteria to be considered whose we can present as the content, user profile, popularity, quality …[2]

Despite its great advantages, centralized IRS presents some limitations. Firstly, the treatment of an immense quantity of documents as well as its organization is becoming the primary impediment to get relevant information. Secondly, the real human opinions about documents are not available in these centralized systems. Several architectures have been used; we note the client/Server, peer to peer and the distributed architecture[5]. Each model uses its own method to rank results. Therefore, we can privilege different formula for the relevance calculation. So, the remaining questions are: what is the best method to have relevant documents? What criteria should be considered? And what is the best implementation to have an efficient information research system.

One of the most difficult problems is the modeling of an efficient system with several documents, topics, queries and users. In this paper, we will try to givea just modeling for the relevance calculation in the distributed collaborative information retrieval systems. Also, we will present different levels of relevance and there combination.

III. RELATED WORKS

Unlike a centralized system,a distributed system mayparallelizetasks and contribute to a large number of machinesin the overall treatment process.The collaborationseveral agents is one of the goals of SRI, several researchers have demonstrated the superiority of collaborative research against individualresearch. In this stage, several works have used the collaborative systems in the knowledge management and the information retrieval.

A. Task-based knowledge support

In the operations and management activities of enterprises,Duen-Ren Liu and I-Chin Wu[6] introduced a Collaborative relevance assessment for task-based knowledge support. They proposed a new task-relevance assessment approach that evaluates the relevance of previous tasks in order to construct a task profile for the current task. The approach helps knowledge workers assess the relevance of previous tasks through linguistic evaluation and the collaboration of knowledge workers. In addition, applying relevance assessment to a large number of tasks may create an excessive burden for workers. Thus, they proposed a novel two-phase relevance assessment method to help workers conduct relevance assessment effectively. Furthermore, a modified relevance feedback technique, which is integrated with the task-relevance assessment method, is employed to derive the task profile for the task-at-hand. Consequently, task-based knowledge support can be enabled to provide knowledge workers with task-relevant information based on task profiles. Empirical experiments demonstrate that the proposed approach models workers' task-needs effectively and helps provide task-relevant knowledge.[7]

B. Collaborative filtering

Based on the classic probability ranking principle, Jun Wang and Arjen P. de Vries [8]proposed a probabilistic user-item relevance model. Under this formal model, they show that user-based and item-based approaches are only two different factorizations with different independence assumptions.Moreover, they show that smoothing is an important aspect to estimate the parameters of the models due to data sparsity. By adding linear interpolation smoothing, the proposed model gives a probabilistic justification of using TF×IDFlike item ranking in collaborative filtering. In this stage, they proposed to apply the probabilistic framework developed for text retrieval to logbased collaborative filtering. They considered the following formal setting. The information that has to be filtered, e.g., images, movies or audio files, is represented as a set of items. They introduced discrete random variablesU$\in \{u_1,...,u_K\}$ and I$\in \{i_1,...,i_M\}$to represent a user and an item in the collection, respectively.*K* is the number of users while *M* is the number of items in the collection. Let L_{uk}denote a user profile list for user $uk \in U$. $L_{uk}$is a set of items that user *uk* has previously shown interest in. $Lu_k(im)$=1 (or $i_m \in L_{uk}$) indicates that item $I_m \in I$, is in the list while $L_{uk}(im) = 0$ (or $I_m \in / L_{uk}$) otherwise. The number of items in the list is denoted as $|L_{uk}|$.

The purpose of log-based collaborative filtering is to rank the relevance of a target item to a user. This could be represented by the retrieval status -*874/5value (RSV) of a target item towards a user, denoted as: $RSV_{uk}(i_m)$.[9]

C. Synchronous Collaborative Information Retrieval with Relevance Feedback

Colum Foley, Alan F. Smeaton and Hyowon Lee[10] are interested in Synchronous Collaborative Information Retrieval that supports 'same-time different-place' collaboration. Their eventual goal was to incorporate developed techniques into a co-located collaborative search system ('same-time same-place') called Fischlar-DiamondTouch, which they have developed and describedelsewhere. In their works, they demonstrated a system forcollaborative searching through video where users shared atouch-sensitive tabletop interface to a search engine and userscommunicated and collaborated in a face-to-face manner inorder to solve a shared information need. Through developingthis system, they have appreciated the need to support collaboration within the underlying IR system itself and not justas part of the interface. The CATS collaborative grouprecommender system is another CSCW (Computer Supported Cooperative Work) system supporting same-time same-place' collaboration[11]. Ski-holiday critiques from multiple users are leveraged and destinations arerecommended based on both an individual's preferences andthe group's preferences. The key objective within the CATsSystem is allowing each user to see which ski-packages suit both their own preferences and those of the groups.

In order to support effective Synchronous Collaborative IR, it is important to allow a search task be divided amongst co-searchers and enable the sharing of knowledge across group members[11]. In the previous work[10], it was proposed an environment whereby the search engine decides on how to divide the task amongst collaborators by showing only novel information to each co-searcher. The sharing of expertise and transfer of knowledge is achievedthrough the IR process of Relevance Feedback. By providing support for both task division and knowledge transfer within the framework of underlying IR system side, it is possible to develop a more effective Synchronous Collaborative Information Retrieval environment.

D. *Expanding queries with collaborative annotation*

Christina Lioma presented a technique for pseudo relevance feedback [12], which expands queries with semantic annotation found in freely available collaborative tagging systems. They hypothesized that collaborative tags can represent semantic information that might be used to enrich queries, and hence enhance retrieval performance. They experimented with three different techniques of enriching queries with collaborative semantic annotation: based on individual terms, based on phrases, and based on whole queries. They also experimented with the number of terms used for expansion, ranging it between 1 and 10. Out of the three techniques, the ones conveying context (phrase-based and query-based) behaved generally similarly; better performance was associated with the query-based technique and fewer expansion terms. Experiments with 36 Web queries showed no significant difference in retrieval performance between the original queries and the expanded queries. Some queries benefited from the developed technique, yet others did not; overall results are inconclusive. Collaborative semantic annotation[13] seems to be broader than or quite general with respect to the user query, suggesting that perhaps better applications for it would be in aiding user interaction, facilitating browsing and serendipitous search, or clustering documents, for instance. Further experimentation is needed in this direction, and particularly with regards to the selection of the most appropriate terms (e.g. by looking at their term statistics, or comparing their distribution in a general document collection to the distribution of query terms in the same collection, to identify discriminative terms).

IV. CONTRIBUTION

A. *Presentation*

In this contribution, we will present a new modeling for the collaborative relevance in the distributed systems. Firstly, we will define the architecture of our system. Then, we will explain the running of each included agent. Finally, we will present a final formula to calculate relevance.

Unlike a centralized system, a distributed system can parallelize tasks and contribute to a large number of machines in the overall treatment process [14]. Our idea is based on the collaboration of users by considering the machines as servers whose their role is to perform the tasks themselves that their interest. This autonomy allows sharing effort between all machines in the group. However, proper management of interactions remains essential.similar work is already introduced by *Colum Foley*[10]. But it was presented without global relevance formula. And it was focused to the architecture of system.
In order to fill up the previous works, we will introduce an improvement of architecture and its principle.

B. *Architecture of the Collaborative Distributed System for Relevance(CDSR)*

In this architecture, we consider all of the Web surfers as system‘s Agents. Each one is an active element of the Network.It can request needed information and in the same time give a relevant documents. The Network is structured as a various clusters that each one links many computers that share similartopics. Certainly, the computers contain all of needed content feedback which reflects the approximate user profile [15].
The user profile is the basic element to define clusters. We can see for example a cluster of sport, Art or Technology. The following diagram explains the architecture of the CDSR.

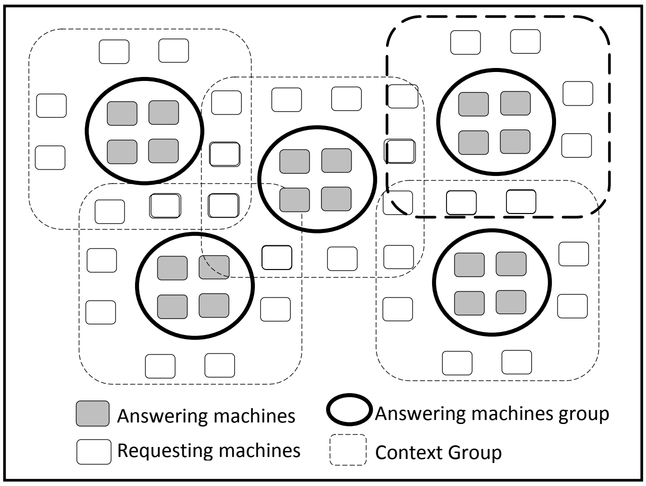


Fig 1: Virtual architecture of the collaborative distributed system of Relevance (CDSR)

In the figure above, the network is organized by clusters. Every one works as an autonomic entity. It contains several machines whose role is to search information or give needed relevant documents.

The machines in the circle are elected by their neighborsin order to bethe relevant responders in the cluster. Therefore, each responder is affiliatedin one or more areas. It's similar as a forum which brings together users who have the same interests. Below, we explain the role of each component of the architecture:

1) *Answering machine:* It is a host which contains the relevant information; it is one of the most active hosts in its cluster. Its activity is defined by its degree of attendance in the domain's cluster among all hosts in the system. Therefore, it's the trusted source of information.

2) *The requesting machines:* this is a set of machines which require information. Theysend requests to the answering machines and collect document in order to calculate relevance. The requesting machines havethree principal functions: send request, receive and sort results.

C. The kinds of local relevance :

1) The Relevance out-document (Rout)

To use the system, we must install a web profiler. This program listens to the visited documents and save information about them. The registering xml file contains the date of last visit, the frequency and the total duration of visits. We use these factors to calculate the relevance out-document (R_{out}).

We notice that the document becomes very interesting when its frequency of visit is high, its total duration of visits is long and its *desertion duration*is low. More the*desertion duration* is highmore the document is old. We consider that the *desertion duration*is inverse proportional with relevance contrary to the frequency and total delay of visits. Therefore, we present the following formula for the Relevance out-document (R_{out}):

$$R_{out}(D_{/user}) = \frac{Frequency * TotalDelay}{e^{d/k}}$$

With: d: *Duration ofdesertion = Now – date of last visit*

and k: factor of flattening.

The duration of desertion gives us an information about the novelty of document. We use this factor because highlight the newest documents and give them more chance to be referenced in first time.

For each visited document, the Web Profiler updates these three values in order to recalculate the *Relevance out-document.* It creates an XML file to register them and indexes documents whose R_{out}exceeds a defined sill.

Finally, for each machine wehave a set of visited and indexed documents, and for each indexed document, we have a set of weighted words. In addition, each visited document have a *Relevance out-document*, which is calculated by the defined formula above.

2) The Relevance in-document (Rin)

This Relevance concerns the content of document. It is related to the frequency of apparition of words [16], its position (title, legend, menu, …) and its format. We calculate the weight of words by the formula below:

$$R_{in}(w_{k_{/D}}) = \frac{\sum_{i=1}^{n} P(w_{ki}) \times F(w_{ki})}{\sum_{l=1}^{t}\sum_{i=1}^{m} P(w_{li}) \times F(w_{li})}$$

W_{ki}: The i^{nd} occurrence of the word W_k
W_{li}: The i^{nd} occurrence of the word W_l
n: number of occurrences of W_k
m:number of occurrences of Wℓ
t: number of words in the document
P: position value
F: format value

The values of P and F are defined in the table below:

Position	Value
First title	10
Second title	8
Third title	6
Legend	4
Element of List	2

Table of position weighting

Format	Value
Bold, Italic, highlighted	10
Big size	8
Colored	6

Table of format weighting

Now, each word have a weight. We can also create an inverse index of words. It contains a set of words accompanied with its container documents [17]. The index is saved in a XML file that the structure is described by the following structure:

	D_1(Rout)	D_2(Rout)	D_3(Rout)	…	D_n(Rout)
W_1(Rp)	Rin	Rin	Rin	Rin	Rin
W_2(Rp)	Rin	Rin	Rin	Rin	Rin
W_3(Rp)	Rin	Rin	Rin	Rin	Rin
. . .					
W_n(Rp)	Rin	Rin	Rin	Rin	Rin

Relevance matrix in local machine

In the table above, we see three kinds of relevance: The relevance out document, the relevance in document and the relevance of profile. The following paragraph explain how we calculate this last relevance.

3) *The relevance of user profile (Rp)*

In the web history, words can be cited in several documents with different R_{in}. Therefore, the word is present once in the invers index. Different R_{in} are reduced to one Relevance that is named “The relevance of user profile”. In fact, the difference between R_{in} and R_p is defined by the related object, Document or User profile. When the R_{in} defines the weight of word in the document, the R_p define the weight of word in all documents indexed by user. Therefore, it defines the weight of word in the user profile.The more weighted words make the profile. The following formula calculate the R_p:

$$R_p(user_{/w_k}) = \frac{\sum_{i=1}^{n} R_{in}(w_{k_{/D_i}}) \times R_{out}(D_{i_{/user}})}{\sum_{i=1}^{n} R_{out}\,(D_{i_{/user}})}$$

This formula presents simply the average of R_{in} showing the R_{out} as a coefficient.

4) *The local relevance*

The local relevance is presents the similarity between query and document regarding a user profile. Unlike the classic similarity, we have proceeded by a broad calculation that covers three factors: query, document and user profile. Therefore, we consider the simple similarity of Jaccard in where we inject the R_{out}.

For each document, we have a weight of words. This weight is exactly the R_{in}. Now, the R_{in} is multiplied by the R_{out} to boost the weight in order to cover the interest of user in the relevance calculation. Therefore, the local weight of word regarding the document container is given by the following formula:

$$R_L(w_{/D}) = R_{in}(w_{/D}) \times R_{out}(D_{/user})$$

This means the local relevance of documents for a query with a single word. For the queries multi-words, we use the Jaccard formula [18]to calculate similarity between query’s words and document’s words (weighted by R_L).

$$R_L(\vec{Q},\vec{D}) = \frac{\sum R_{iQ} \times R_L(w_{/D})}{\sum R_{iQ} + \sum R_L(w_{/D}) - \sum R_{iQ} \times R_L(w_{/D})}$$

With: R_{iQ}:The weighting of the i^{nd} word in the query’s vector.

and $\mathbf{R_L}(\boldsymbol{w_{/D}})$: The weighting of the i^{nd} word in the document’s vector.

Now, the web profiler program can sort documents by R_L against a query. Therefore, user have possibility to search and order its own documents implicitly by relevance. The following figure recaps the different local relevance and the highlighting objects.

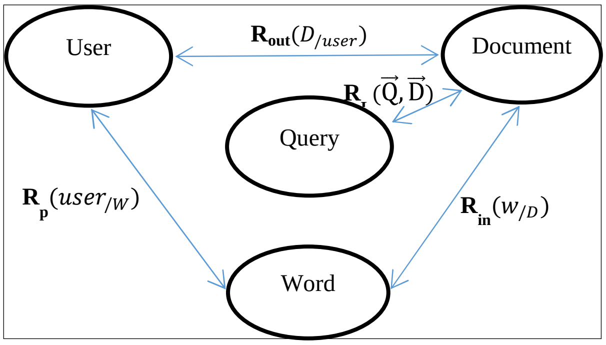

User
$\mathbf{R_{out}}(D_{/user})$
Document
$\mathbf{R_L}(\vec{Q},\vec{D})$
Query
$\mathbf{R_p}(user_{/W})$
$\mathbf{R_{in}}(w_{/D})$
Word

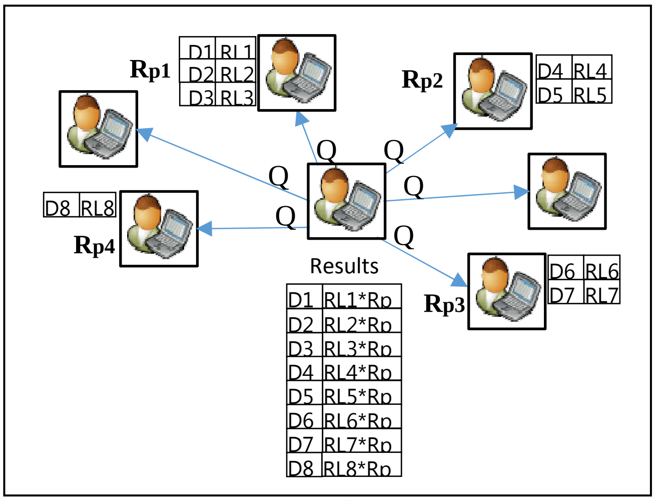
Fig 2: System objects and relevance

Finally, we consider that the words are nodes of graph where the user is an affiliated element to one or more nodes. The Rp is the degree of attachment to the node. The following figure shows relation between the user and words.

D. *The collaborative Relevance*

The collaborative relevance define the satisfaction of the most users of system. More the users are satisfied about a document more its relevance is perfect. In the local machine, the user have possibility to sort its documents using the local relevance compared to query. The local machine give its own decision about a search task. To generalize, we consider that we have several machines. Each one give its decision about query. We take the different decisions and we combine them in order to have a collaborative decision. This is the principal idea to sort results in a distributed information retrieval.

To retrieve information, the requesting machine send query to the answering machines. The last, search inside in the index and match query’s words to get documents. In the fig 3, we illustrate the calculation model of the collaborative relevance.



D1	RL1*Rp
D2	RL2*Rp
D3	RL3*Rp
D4	RL4*Rp
D5	RL5*Rp
D6	RL6*Rp
D7	RL7*Rp
D8	RL8*Rp

Fig 3: The calculation model of the collaborative relevance

In the figure above, we notice that each answering machine injects its Rp into the local relevance.The Rp($user_{/w_k}$)means the interest of user abouta word. The users having a high Rp compared to words of query, give documents more relevant, because more a user is very interested by a concept more its document container is important for him. The calculation of Rp defends this idea, and gives us a ranking of users in one or more topics. The ranked users have more possibilities to contain relevant information. We can also define some classified users as relevant responders. For this, we use a sill of Rp for the pair of (user, word). Users whose Rp is upper than defined sill, will be selected to be the relevant and expert users for the current search.

Finally, the collaborative relevance is depending to the weighting of user about query; the weighting of document in each local machine and the weighting of query inside each the document. The calculation of the collaborative relevance is given by the following formula:

$$R_c(Q,D) = \frac{\sum_{k=1}^{n} R_p(User_{k_{/Q}}) \times R_L(\vec{\mathrm{Q}},\vec{\mathrm{D}})_{/User_k}}{\sum_{k=1}^{n} R_p(User_{k_{/Q}})}$$

In this formula, all of users give their documents rated by R_L. Each R_L is multiplied by the user weight (as coefficient) and finally divided by the sum of coefficients. The final sorting of documents is based on the Rc.

V. Conclusion and prospects

In this paper, we have presented a new method for ranking results in the meta-search engine. We have started by a definition of the related works and we have presented our architecture of distributed system. In our contribution, we have defined our kinds of relevance and we have given the formula of calculation of each kind. In addition, we have presented the idea of the collaborative relevance and we have given the different relevance's calculation in the system. We have introduced the importance of the local ranking and the weight of each user to have a relevant result.

In the future, we will improve the system by giving more of precision. The ambiguity of terms is one of several problems to resolve. We think to use the TALN methods and ontologies to achieve a semantic search. In order to cover queries, documents and users, we think to use the contextual graphs to link all of system's elements and create a general ontology for the collaborative-search systems.

References

[1] E. G. Ularu, F. C. Puican, A. Apostu and M. Velicanu, "Perspectives on Big Data and Big Data Analytics," Database Systems Journal, vol. 3, no. 4, 2012.

[2] P, Patil Swati; B.V, Pawar; S, Patil Ajay, "Search Engine Optimization: A Study," Research Journal of Computer and Information Technology Sciences, vol. 1, no. 1, pp. 10-13, 2013.

[3] A. Enaanai and S. A. Doukkali, "An hybrid approach to calculate remevance in the meta-search engines," International Journal of Science and Advanced Technology, vol. 13, no. 2, 2012.

[4] M. Zolghadri-Jahromi and M. Valizadeh, "A proposed query-sensitive similarity measure for information retrieval," Iranian Journal of Science & Technology, Transaction B, Engineering, vol. 30, no. 2, pp. 171-180, 2006.

[5] A. Kermarrec and F. Taïani, "Want to scale in centralized systems? Think P2P," Journal of Internet Services and Applications, vol. 6, no. 16, 2015.

[6] D.-R. Liu and I.-C. Wu, "Collaborative relevance assessment for task-based knowledge support," Decision Support Systems, vol. 44, no. 2, pp. 524-543, 2008.

[7] C.-K. Ke and D.-R. Liu, "context-based knowledge support for problemsolving," International Journal of Innovative Computing, Information and Control, vol. 7, no. 7, p. 3615–3631, 2011.

[8] J. Wang and P. d. V. Arjen, "Unified relevance models for rating prediction in collaborative filtering," ACM Transactions on Information Systems (TOIS), vol. 26, no. 3, 2008.

[9] J. Wang and P. d. V. Arjen, "Probabilistic relevance ranking for collaborative filtering," Information Retrieval, vol. 11, no. 6, pp. 477-497, 2008.

[10] C. Foley, A. F. Smeaton et H. Lee, «Synchronous Collaborative Information Retrieval with Relevance Feedback,» 2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing, IEEE Explore, pp. 1-4, 2006.

[11] K. Mccarthy, L. Coyle, L. Mcginty, B. Smyth et P. Nixon, «Cats: A synchronous approach to collaborative group recommendation,» Proceedings of the International FLAIRS Conference, 2006.

[12] C. Lioma, M.-F. Moens and L. Azzopardi, "Collaborative annotation for pseudo relevance feedback," Proceedings of exploiting semantic annotation in information retrieval, pp. 25-35, 2008.

[13] D. E. Zomahoun, "collaborative semantic annotation of images: ontology-based model," Signal & Image Processing : An International Journal (SIPIJ), vol. 4, no. 6, 2013.

[14] A. W. Rohankar, Mrinal K. Naskar and Amitava Mukherjee, "SWiFiNet : A Task Distributed System Architecture for WSN" International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Selected Papers from International Conference.

[15] H. Naderi, B. Rumpler and J.-M. Pinon, "An Efficient Collaborative Information Retrieval System by Incorporating the User Profile," 4th International Workshop, AMR 2006, Geneva, Switzerland, July 27-28, 2006, Revised Selected Papers, pp. 247-257, 2006.

[16] H. Wu, R. Luk, R. Wong and K. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," ACM Transactions on Information Systems, vol. 26 , no. 3, 2008.

[17] F. Boubekeur and W. Azzoug, "concept-based indexing in text information retrieval," International Journal of Computer Science & Information Technology (IJCSIT), vol. 5, no. 1, 2013.

[18] L. Hamers, Y. Hemeryck, G. Herweyers and M. Janssen, "Similarity measures in scientometric research: The Jaccard index versus Salton's cosine formula," Information Processing & Management, vol. 25, no. 3, pp. 315-318, 1989.