Learning and Modelling User Interests using User Feedback : a Novel Approach

Tarek Alloui
MISC Laboratory,
Dept of Computer Science and its Applications,
Faculty of NTIC,
University Constantine 2 Abdelhamid Mehri
Constantine, Algeria
tarek.alloui@univ-constantine2.dz

Imane Boussebough
LIRE Laboratory,
Dept of Software Technology and Information Systems
Faculty of NTIC,
University Constantine 2 Abdelhamid Mehri
Constantine, Algeria
iboussebough@gmail.com

Allaoua Chaoui
MISC Laboratory,
Dept of Computer Science and its Applications,
Faculty of NTIC, University Constantine 2 Abdelhamid Mehri
Constantine, Algeria
a_chaoui2001@yahoo.com

***Abstract*—User profiles and interests have become essential for personalizing information search and retrieval. Indeed, traditional Information Retrieval Systems (IRS) don't integrate the user in the search process. Also, users do not always find what they need after a single query. Instead, they often issue multiple queries, incorporating what they learned from the previous results to iterate and refine how they express their information needs. So we rely on this process to learn the user information needs without asking him explicitly. This is achieved by capturing his judgments on the retrieved results. We consider also, in the construction of the user interests, what he is looking for and what the user doesn't want to find in the future results to build interests that best match his information needs.**

Keywords— user interests; information retrieval systems; user feedback; user information needs

I. INTRODUCTION

Information retrieval is a computer science branch whose main goal is acquiring, organizing, storing, retrieving and selecting information for the user. From the users' perspective, accessing information can be performed deliberately through an Information Retrieval System (IRS) using queries as formal statements for his information needs.

But over the last century, IR has faced new challenges. The wide popularization of personal computers, the development of electronic media with growing capacities, the Internet revolution which is considered as the largest source of information in the world and the ubiquity of mobile networks nowadays allow users worldwide to access a huge mass of information. Therefore, obtaining the right information becomes crucial, and finding the relevant information suitable to the users' needs becomes more and more important and difficult.

In addition, there isn't a model for all situations. Indeed, there is a real divergence between IR and its methods in representing user information needs with his own query. So, if the same query is expressed by two users whose information needs are different, it will return the same results. Moreover, the user formulates his query using few words which lead to incomplete and imprecise specifications of real informational needs.

So, with the user playing a major role in the search process, it is important to personalize the IR system according to his preferences, interests and even his individual search environment in order to increase the relevance of returned results.

Representing the user by a structure or a model is not an easy task, but it is an essential element that consequently influences the performances of IR systems. The difficulty remains in the steps of user profile construction, the faithful representation of his interests, their evolution and their exploitation in the search process in order to improve the retrieved results.

Since the reflection process in this area is recent, few standards are established. But several approaches in personalized IR focused on the profile construction in order to identify user information needs and thus improve the returned results.

One of the most popular and widely used strategies is Relevance Feedback (RF). It can be classified into three

ISBN: 978-1-4673-6832-2

categories: Explicit Relevance Feedback (ERF), Implicit Relevance Feedback (IRF) and Pseudo Relevance Feedback (PRF). Many approaches worked on building user profile or interests explicitly [1][2] or implicitly [3][4][5]. Indeed, implicit feedback has become the most used technique in understanding user information needs [8][9][10]. But recently, with the increase of feedback information, explicit feedback has been a focus for researchers [11][12]. In fact, even if implicit feedback such as clickthroughs, scrolling or time spent on documents have given good results in understanding user behavior [11], little feedback information is known from these kind of interactions. Therefore, explicit feedback gives much more relevant and reliable information about what the user is really looking for and what he is not looking for by identifying relevant and irrelevant results. Also, the user is and will remain the best judge of the results relevance. In [6], authors use only relevant documents to build user interests. In our work, we rely also on irrelevant documents to build user interests.

In this paper we present a new approach for building user interests using only his interactions with our system without asking them explicitly. The novelty introduced in this approach compared to other existing works is that we try to learn from the users what is relevant or not to their information needs. We rely on Rocchio's work [7] to determine user interests by considering the documents he judges to be relevant or irrelevant for his needs.

II. The Proposed Approach

The proposed system's main objective is the personalization of information retrieval, i.e., we want to build user interests in order to use them in future works for improving his future queries by relevance feedback taken from the user interests already built. To build relevant interests, we set up a learning process from user experiences. These experiences are mainly based on user questioning and interactions with search engine through our system.

In a classical Web search scenario, users do not always find what they seek after a single query. Instead, they always try multiple queries, by incorporating and/or modifying the first query with new keywords in order to iterate and refine how they express their information needs. So, we chose to take advantage of this process to learn what the user information needs are and therefore build his interests.

The aim of this phase is to learn which information the user is looking for, what are the documents he considers as relevant or irrelevant for his information needs in order to establish reliable user interests that can be used later in a query reformulation process in order to retrieve more relevant documents that best matches user's needs.

In the first iteration, the user enters a query through the system interface using his own keywords. Then, the system submits this query to the search engine, retrieves the results and presents them to the user as they are returned from the search engine. Thereafter, the user will examine the results and will judge each result by giving it a score between 1 and 5 according to the result relevancy. The scoring principle is as follows:

Table 1. The relevance levels of the retrieved documents

Score	Relevance
1	Not relevant
2	Not very relevant
3	Potentially relevant
4	Relevant
5	Very relevant

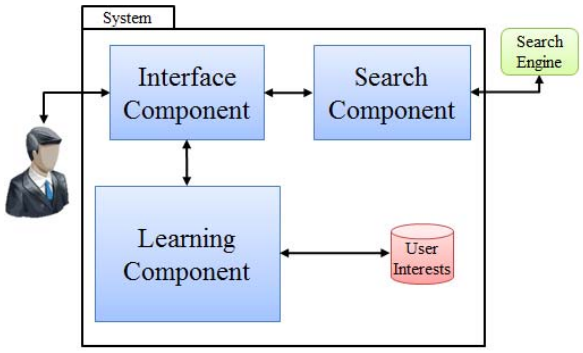

Figure 1. Architecture of the proposed system

This rating system is inspired from what is done in recommendation systems. As said before, we deliberately chose the explicit feedback because the user is the final judge of documents relevancy and this way the system can rely on his judgments to build and effective and reflective user interests.

The results scored in {1,2} will be considered as irrelevant, and those scored in {4,5} will be considered as relevant. The results scored "3" will be considered as ambiguous, i.e., the user still hesitates on their relevancy.

Once these judgements got by the system, it will analyze them to retrieve keywords and starts building user interests. The principle of this process is summarized in this algorithm:

```
Algorithm
While (Building Phase not finished) Do
      Step 1: User enters / modifies his
      query
      Step 2: System submit user query and
      retrieve results from search engine
      Step 3: User consults results and
      gives them scores according to (Table
      1)
      Step 4: System gets user judgements
      on retrieved results
      For (Each retrieved Result) Do
       If (Result is scored in {1,2}) Then
         Result is considered as irrelevant
       If (Result is scored in {4,5}) Then
         Result is considered as relevant
       If (Result is scored "3") Then
         Result is considered as ambiguous
    EndFor
EndWhile
```

A. Representation of user interests

The keywords extracted from documents considered by the user will allow us to build his interests. So we proposed a model based on sets defined as follows:

- K_r: set of relevant keywords.
- K_{ir}: set of irrelevant keywords.
- K_{tmp}: set of ambiguous keywords.
- K_u: set of user keywords.

B. Principle of user interests construction

After extracting the keywords of each document, the keywords of relevant documents will be added to K_r set and those of irrelevant documents will be added to K_{ir} set. Keywords of documents scored "3" will be added to K_{tmp} set.

The definition of our model leads to the following intersections:

- $K_u \cap K_r \neq \emptyset$: the system can found user keywords in relevant documents.
- ($K_u \cap K_{ir} = \emptyset$) and ($K_u \cap K_{tmp} = \emptyset$): user keywords are deleted from K_{ir} and K_{tmp} in order to preserve the user information needs.
- ($K_r \cap K_{ir} = \emptyset$) and ($K_r \cap K_{tmp} = \emptyset$) and ($K_{ir} \cap K_{tmp} = \emptyset$): the K_{tmp} set represents the intersection that should appear between K_r and K_{ir}, and during the learning process, the system will assign K_{tmp} keywords to the first or second set.

In addition, during the construction phase, our system may encounter a contradictory situation where a word was found in the previous iterations as irrelevant become relevant. This keyword should be moved to the K_{tmp} set until the system learns enough from the user about it to classify it in K_r or K_{ir} set. To do this, we associate to each keyword of the 3 sets (K_r, K_{ir}, K_{tmp}) two judgment counters C_r and C_{ir}.

1) Definition 1

The relevance counter, noted "C_r", is an integer associated to each keyword. It will be incremented each time the system found this word in a relevant document.

2) Definition 2

The irrelevance counter, noted "C_{ir}", is an integer associated to each keyword. It will be incremented each time the system found this word in an irrelevant document.

3) Definition 3

The relevancy rate, noted "$R_{relevancy}$", represents the relevancy of a keyword to user information needs. It will allow our system to decide in which set the keyword will be added. This rate is calculated as follows:

$$R_{relevancy} = C_r / C_{ir}$$

For each iteration, the K_r, K_{ir} and K_{tmp} sets will be updated as follows:

```
Algorithm
For (Each keyword of documents scored 5)
Do
  If ((keyword in Kr) or (keyword in
    Ktmp)) Then
      - Cr ← Cr + 1
  If (keyword not in Kr) Then
      - Add it to Kr set
      - Cr ← Cr + 1
EndFor
For (Each keyword of documents scored 4)
Do
  If ((keyword in Kr) or (keyword in
    Ktmp)) Then
      - Cr ← Cr + 0.5
  If (keyword not in Kr) Then
      - Add it to Kr set
      - Cr ← Cr + 0.5
EndFor
For (Each keyword of documents scored 1)
Do
  If ((keyword in Kir) or (keyword in
    Ktmp)) Then
      - Cir ← Cir + 1
  If (keyword not in Kir) Then
      - Add it to Kir set
      - Cir ← Cir + 1
EndFor
For (Each keyword of documents scored 2)
Do
  If ((keyword in Kir) or (keyword in
    Ktmp))Then
      - Cir ← Cir + 0.5
  If (keyword not in Kir) Then
      - Add it to Kir set
      - Cir ← Cir + 0.5
EndFor
```

We fixed the value "1" as a threshold of the relevancy rate $R_{relevancy}$. At the end of each iteration of the construction phase, the system will calculate the relevancy rate $R_{relevancy}$ for each keywords and proceeds as follows:

```
Algorithm
For (each keyword in Kr, Kir and Ktmp) Do
  If (Rrelevancy > 1) Then
    If (keyword not in Kr) Then
      Add it to Kr set
  If (Rrelevancy < 1) Then
    If (keyword not in Kir) Then
      Add it to Kir set
  If (Rrelevancy = 1) Then
    If (keyword not in Ktmp) Then
      Add it to Ktmp set
EndFor
```

$$\alpha + \beta = \chi \quad (1) \qquad (1)$$

At the end of the learning process, the system will set up two sets:

- The K_r set: it will contain all keywords taken from documents considered as relevant by the user and those that best matches user information needs. It will be used later to enrich the user query with more information.
- The K_{ir} set: it will contain all keywords extracted from documents considered as irrelevant. This will be used later to enrich the user query to indicate what the user would not find in the future results.

This sets idea allows us to define our representation model of user interests. These sets can be used in a user queries reformulation approach to improve the rate of relevant documents.

III. Experimental Results and Discussion

To test the feasibility and effectiveness of our approach, we implemented a prototype of the system using Google as a search engine. Instead of testing our approach on document collection, we preferred testing it directly on a large scale collection, i.e., the Web, to build real users interests in order to more appreciate the approach effectiveness. While testing our system on building different users interests, we noticed that through only three iterations the system converges quickly toward the real user information needs.

To illustrate these results, we built 5 different user interests. Using these user interests, we calculated an average number of K_r keywords and an average number of K_{ir} keywords on 10 iterations of the construction phase.

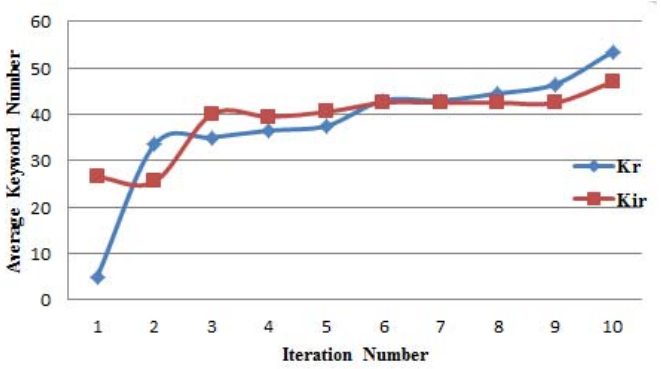

Figure 2. Evolution of the average number of K_r and K_{ir} keywords through iterations

We can see from **Figure 2** that the number of keywords in both K_r and K_{ir} sets become significant during the three first iterations and become more or less stable in the next iterations. Also, this can be verified in **Figure 3** and **Figure 4**, the number of new keywords added to K_r and K_{ir} sets is bigger in the three first iterations of the construction phase and becomes almost near to "0" during the other iterations.

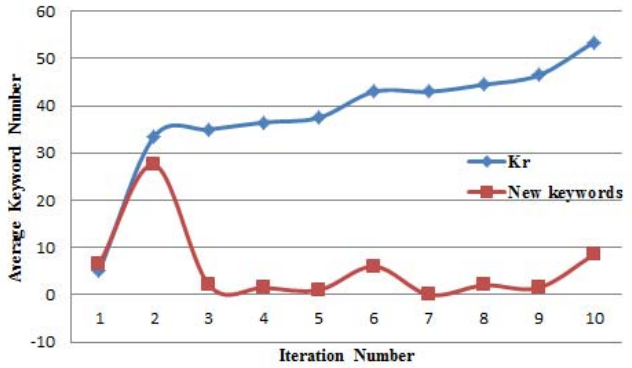

Figure 3. Evolution of the average number of new keywords added to K_r set

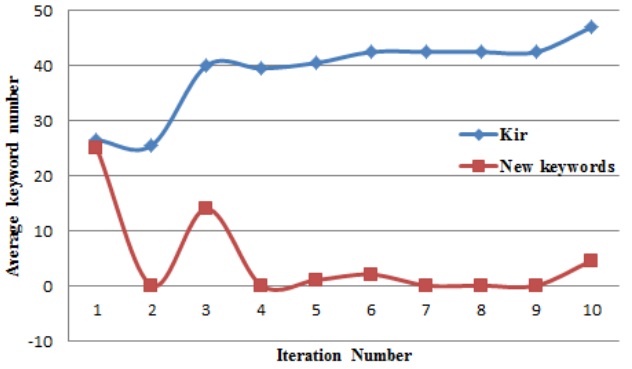

Figure 4. Evolution of the average number of new keywords added to K_{ir} set

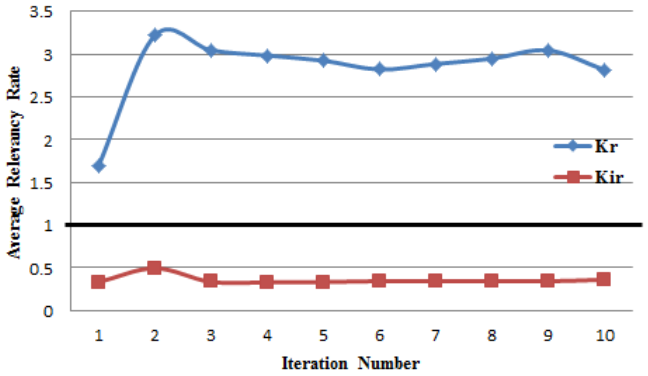

Figure 5. Evolution of the average $R_{relevancy}$ of K_r and K_{ir} keywords

Now for the relevancy rate $R_{relevancy}$, in **Figure 5** we notice that for K_r keywords, the average rate increases quickly above the threshold "1" in the three first iterations and become regular in the next iterations. This is due to the convergence of user interests toward his real information needs.

We can observe the same results with the relevancy rate of K_{ir} keywords, i.e., it becomes stable under the threshold "1" after the three first iterations because the system has converged toward what the user doesn't want to find in the future results.

So, the system can quickly, without annoying the user with a big number of iterations (it doesn't go over three iterations), build meaningful and reliable user interests which can be used

in a query reformulation approach, to get more relevant documents that best matches user information needs.

IV. Conclusion and Perspectives

In this paper, we presented a novel approach of quickly building user interests without asking the user to enter them explicitly. Our main goal is to build user interests that well reflect his information needs in order to use them in a future work for user queries reformulation and thus improve the future results.

And to achieve this goal, we proposed a system with a construction phase where the system learns the user information needs from his judgments on the retrieved results. During this phase, the system will build two sets of keywords extracted from the relevant and irrelevant documents. These two sets will represent user interests.

We can see from our experimental results that the system can built quickly and efficiently, with only three iterations of the construction phase, a meaningful and useful user interests. Using relevance feedback from user interests built in this phase, we can surely be able to reformulate user's future queries and thus get more relevant documents to his information needs.

As said before, we plan to use the results of this construction phase to reformulate by relevance feedback user queries. We propose to integrate into our system multiple search engines querying to present more results to the user and thus enrich more his interests. We also plan to work on the temporal aspect of user interests as a short or long term profile.

References

[1] Z. Z. Nick and P. Themis, Web search using a genetic algorithm, *IEEE Internet Computing*, p. 18-26, March-April 2001.

[2] Z. Ma, G. Pant, and O. R. Liu Sheng, Interest-based Personalized Search, *ACM Transactions on Information Systems*, vol. 25, No. 1, Article No. 5, February 2007.

[3] H. R. Kim and P. K. Chan, IUI '03: Learning implicit user interest hierarchy for context in personalization, *Proceedings of the 8th international conference on Intelligent user interfaces*, ACM, New York, NY, USA, p. 101-108, 2003.

[4] F. Liu, C. Yu, and W. Meng, Personalized Web Search For Improving Retrieval Effectiveness, *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, No. 1, p. 28-40, 2004.

[5] X. Shen, B. Tan, and C. Zhai, Implicit user modeling for personalized search, *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM, New York, NY, USA, p. 824-831, 2005.

[6] L. Tamine, M. Boughanem, and W. N. Zemirli, Personalized document ranking: Exploiting evidence from multiple user interests for profiling and retrieval, *Journal of Digital Information Management*, vol. 6, n° 5, p. 354-365, October, 2008.

[7] J.J. Rocchio, Relevance feedback in information retrieval, *The Smart retrieval system - experiments in automatic document processing*, Prentice Hall Inc., p. 313-323, 1971.

[8] D. Kelly and J. Teevan, Implicit Feedback for Inferring User Preference: A Bibliography, *ACM SIGIR Forum*, vol. 37, No. 2, p. 18-28,2003.

[9] G. Buscher, J. Gwizdka, J. Teevan, N. J. Belkin, R. Bierig, L. Van Elst, and J. Jose, SIGIR 2009 workshop on understanding the user- logging and interpreting user interactions in information search and retrieval, *Proceedings of the SIGIR 2009 Workshop,* 2009.

[10] J. Y. Kim, M. Cramer, J. Teevan, and D. Lagun, Understanding how people interact with web search results that change in real-time using implicit feedback, *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, San Francisco, California, USA, p. 2321-2326, 2013.

[11] C. Chen, H. Chunyan, and Y. Xiaojie, Relevance Feedback Fusion via Query Expansion, *WI-IAT '12 Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 3, p. 122-126, 2012.

[12] D. Lagun, A. Sud, R. W. White, P. Bailey, and G. Buscher, Explicit feedback in local search tasks, *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, Dublin, Ireland, p. 1065-1068, 2013.