Integrating user's profile in the query model for Social Information Retrieval

Chahrazed Bouhini, Mathias Géry, Christine Largeron

Integrating user's profile in the query model for Social Information Retrieval

Chahrazed Bouhini*, Mathias Géry*, Christine Largeron*
*Université de Lyon, F-42023, Saint-Étienne, France,
CNRS, UMR 5516, Laboratoire Hubert Curien, F-42000, Saint-Étienne, France
Université de Saint-Étienne, Jean-Monnet, F-42000, Saint-Étienne, France
{ chahrazed.bouhini, mathias.gery, christine.largeron }@univ-st-etienne.fr

***Abstract*—Social Information Retrieval (SIR) exploits the user's social data in order to refine the retrieval, for instance in the case where users with different backgrounds may express different information needs as a same textual query. However, this additional source of information is not supported by the classical IR process. In this article, we propose an approach to generate the user profile from his social data. This generated profile is integrated within a SIR model allowing to personalize the list of documents returned to the user.**

***Keywords*—Social information retrieval model, user-centered IR model, personalized SIR model.**

I. Context

Classical IR systems are based on the keyword search: given a collection of resources and a user information need, they aim to provide relevant resources to the user. Generally, the information need is expressed with a query composed of a few keywords, usually less than three words because few users are able to formulate their needs with complex queries [1]. However, it has been shown that the queries can be ambiguous since users with different needs can use the same query even if they expect different relevant resources [2]. A solution to enhance the IR process, without altering the way that the users specify their requests, is personalized IR which takes into account not only the query but also other information given directly or not by the user [3]. This leads to a Social Information Retrieval (SIR) which attempts to extend classical IR by taking into consideration the user's profile. The user's profile can be integrated in order to refine the query, like in query expansion or during the indexation and the ranking of the documents. In this last case, considered in this article, the ranking of a document depends not only on the matching between the document and the query, but also on the matching between the user's interest and the document. The user's profile can be generated from his social annotations. This profile can also integrate the tags used by the user's neighbors in his social network. In several models, notably inspired by the Vector Space Model, the document, the query and the profile are then described by vectors defined in the same space: the tags. The matching between a query and a document as well as the matching between the user's profile and the document are computed with the cosine similarity. This approach have notably been proposed in ([4], [5], [6], [7], [8]). In this article we propose a framework for personalized IR based on folksonomies and we investigate the way to integrate it in SIR models. Finally, we propose different SIR models based on the well known IR model BM25 [9].

II. Social information retrieval model

Our model is based on the following principles: Firstly, the user's information needs can be represented not only by the query but also by the user's profile. Secondly, as the user's information needs are more complex, it might be interesting to consider the term frequencies, like weighting functions do on the document. Consequently, our approach attempts to combine binary queries with user's profile based on term frequencies. In this work, we propose to study two different strategies: the first one combines the two kind of information at a scoring level, while the second one combines them at a term frequencies level. This last strategy is based on the work of Robertson et *al.*, who have shown, while combining several textual fields that compose a structured document, that the second kind of strategy is theoretically and experimentally better than the first one [10].

We present three SIR models that exploit the user's profile in order to refine the user's query. These models are based on BM25 [9] which computes the score between a document d and a query q according to equation (1) and, they are defined as follows:

- $BM25S(d,u)$ (SIR model): this model returns a ranked list of documents that are relevant for a given user u considering only his profile (u). Thus, q is simply replaced by u in equation (1).
- $BM25S_{ScoreComb}(d,q,u)$ (Combined SIR model): this model returns a ranked list of documents that are relevant for a given user u considering his binary query q combined **at the scoring level** with his profile, using the equation (2).

$$BM25S_{ScoreComb}(d,q,u) = BM25(d,q)+\alpha\times BM25S(d,u) \quad (2)$$

- $BM25S_{FreqComb}(d,q,u)$ (Combined SIR model): this model returns a ranked list of documents that are relevant for a given user u considering his binary query q combined **at the term frequencies level** with his profile, as recommanded by Robertson et *al.* [10], using the equation (3).

$$BM25S_{FreqComb}(d,q,u) = BM25(d,q+\alpha\times u) \quad (3)$$

We propose three variants of each of these SIR models, using the BM25 k_3 parameter in order to set the saturation level of the query term frequencies ($k_3 = 0$: maximum saturation,

$$BM25(d,q) = \sum_{t \in q \cap d} \frac{(k_1+1) \times tf_{d,t}}{k_1 \times \left((b-1) + b \times \left(\frac{dl}{avgdl}\right)\right) + tf_{d,t}} \times \log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \times \frac{(k_3+1) \times tf_{q,t}}{k_3 + tf_{q,t}} \quad (1)$$

TABLE I. EXAMPLE: QUERY, DOCUMENTS AND USER'S PROFILES.

	t_1 = smartphone	t_2 = android
Query q	1	1
Document d_1	1	0
Document d_2	0	1
User Bob	2	1
User $Alice$	1	2

$k_3 = 1000$: no saturation [11] and $k_3 = 8$: balanced saturation [9]).

For instance, the SIR model $BM25S(d,u)$, having the query terms replaced by the user's profile terms ($tf_{u,t}$ instead of $tf_{q,t}$, where $tf_{u,t}$ is the frequency of the term t within the user's profile u), is declined in three variants as follows:

- $BM25S_{bin}(d,u)$ (binary profile): equation 1 with $k_3 = 0$ is equivalent to have a user's profile represented by a binary vector.
- $BM25S_{tf}(d,u)$ (frequencies profile): equation 1 with $k_3 = 1000$ is equivalent to a profile represented by a vector of term frequencies which leads to set the saturation off on $tf_{u,t}$ [11].
- $BM25S_w(d,u)$ (weighted profile): equation 1 with $k_3 = 8$ is equivalent to a user's profile represented by a vector of term weights which corresponds to a moderate saturation of $tf_{u,t}$ [9].

III. ILLUSTRATIVE EXAMPLE

Our system should be able to handle ambiguous queries, i.e. queries having potentially several interpretations representing different information needs. For example, suppose that two users Bob and $Alice$ have the same query q = "smartphone android" (cf. Table I). We consider two documents d_1 and d_2; each document contains one query term, but *smartphone* is more important than *android* in the first document since d_1 contains only *smartphone*, and *android* is more important than *smartphone* in the second one since d_2 contains only *android*. Assuming that the two query terms have the same importance, a classical IR system should estimate that d_1 is equally relevant as d_2 for the query "*smartphone android*". However, depending on the user and his profile, the information need behind this query may focus either on the term *smartphone* or on the term *android*.

Bob is mainly interested in smartphone devices, then his information need is probably centered around smartphones with an opening on Android, and thus the query term *smartphone* should be more important than the query term *android*. On the other hand, $Alice$ is mainly interested by the Android operating system, consequently his information need is probably centered around Android, and thus the query term *android* should be more important than the query term *smartphone*.

Table II shows the scores computed for the documents shown in the Table I, using the IR model and the SIR models that exploit the user's profile. The scores have been computed using usual BM25 parameters values: $b = 0.75$, $k_1 = 1.2$.

TABLE II. IMPACT OF THE USER'S INFORMATIONAL SOCIAL CONTEXT ON THE SCORING PROCESS

	Bob		$Alice$	
	d_1	d_2	d_1	d_2
$BM25_{bin}(d,q)$	1.680	1.680	1.680	1.680
$BM25S_{bin}(d,u)$	1.680	1.680	1.680	1.680
$BM25S_{tf}(d,u)$	3.360	1.680	1.680	3.360
$BM25S_{w}(d,u)$	3.025	1.680	1.680	3.025
$BM25S_{ScoreComb-bin}(d,q,u), \alpha = 0.5$	2.520	2.520	2.520	2.520
$BM25S_{ScoreComb-tf}(d,q,u), \alpha = 0.5$	3.360	2.520	2.520	3.360
$BM25S_{ScoreComb-w}(d,q,u), \alpha = 0.5$	3.192	2.520	2.520	3.192
$BM25S_{FreqComb-bin}(d,q,u), \alpha = 0.5$	2.521	2.521	2.521	2.521
$BM25S_{FreqComb-tf}(d,q,u), \alpha = 0.5$	3.361	2.521	2.521	3.361
$BM25S_{FreqComb-w}(d,q,u), \alpha = 0.5$	3.052	2.386	2.386	3.052

This example shows that taking into account the user's profile allows re-ranking the documents list according to our SIR objectives. The ranking of the documents is different for each user issuing the same query but having different information needs. The personalized SIR models ($BM25S$, $BM25S_{ScoreComb}$ and $BM25S_{FreqComb}$) return d_1 first than d_2 for Bob who is more interested by having d_1 in first position than d_2 and the opposite for $Alice$.

REFERENCES

[1] B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic, "Real life information retrieval: A study of user queries on the web," *SIGIR Forum*, vol. 32, no. 1, pp. 5–17, 1998.

[2] M. Sanderson, "Ambiguous queries: test collections need more sense," in *SIGIR*, 2008, pp. 499–506.

[3] E. Agichtein, E. Brill, and S. T. Dumais, "Improving web search ranking by incorporating user behavior information," in *SIGIR*, 2006, pp. 19–26.

[4] J. Diederich and T. Iofciu, "Finding communities of practice from user profiles based on folksonomies," in *TEL-CoPs06*, 2006.

[5] M.-G. Noll and C. Meinel, "Web search personalization via social bookmarking and tagging," in *The Semantic Web*, 2007, vol. 4825, pp. 367–380.

[6] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu, "Exploring folksonomy for personalized search," in *SIGIR*, 2008, pp. 155–162.

[7] Y. Cai and Q. Li, "Personalized search by tag-based user profile and resource profile in collaborative tagging systems," in *CIKM*, 2010, pp. 969–978.

[8] D. Vallet, I. Cantador, and J. M. Jose, "Personalizing web search with folksonomy-based user and document profiles," in *ECIR*, 2010, pp. 420–431.

[9] S. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne, "Okapi at TREC'4," in *TREC-4*, 1996, pp. 73–96.

[10] S. Robertson, H. Zaragoza, and M. Taylor, "Simple BM25 extension to multiple weighted fields," in *CIKM*, 2004, pp. 42–49.

[11] K. S. Jones, S. Walker, and S. Robertson, "A probabilistic model of information retrieval: development and comparative experiments, part 2," *IPM*, vol. 36, pp. 809–840, 2000.