Using Social Annotations to Enhance Document Representation for Personalized Search

Mohamed Reda Bouadjenek[†*], Hakim Hacid[‡*], Mokrane Bouzeghoub[†], Athena Vakali[§]
[†]PRiSM Laboratory, Versailles University, {mrb,mok}@prism.uvsq.fr
[‡]Sidetrade, 114 Rue Gallieni, 92100 Boulogne-Billancourt, France, hhacid@sidetrade.com
[§]Computer Science Department, Aristotle University of Thessaloniki, Greece, avakali@csd.auth.gr

ABSTRACT

In this paper, we present a contribution to IR modeling. We propose an approach that computes on the fly, a Personalized Social Document Representation (PSDR) of each document per user based on his social activities. The PSDRs are used to rank documents with respect to a query. This approach has been intensively evaluated on a large public dataset, showing significant benefits for personalized search.

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

General Terms: Algorithms, Experimentation.

Keywords: Information Retrieval, Social networks.

1. INTRODUCTION

Nowadays, with the huge amount of available information on the Web, finding relevant information remains harder for end-users as: (i) usually, the user doesn't necessarily know what he is looking for until he finds it, and (ii) even if the user knows what he is looking for, he doesn't always know how to formulate the right query to find it (unless in case of navigational queries [7]). In existing IR systems, queries are usually interpreted and processed using document[1] indexes and/or ontologies, which are hidden to the user. The resulting documents are not necessarily relevant from an end-user perspective, in spite of the ranking provided by the Web search engine.

One way to improve the IR process and reduce the amount of irrelevant documents is to improve the IR model, which is the focus of this work. Modeling in IR consists of two main tasks: (i) the definition of a *conceptual model* to represent documents and queries and (ii) the definition of a ranking function to quantify the similarities among documents and queries. We believe that enhancing the representation of documents with social information while personalizing them is expected to improve web search. Our motivations to improve the IR model are mainly driven by the following observations:

1. "Social contextual summarization" is required due to the fact that with the advent of the social Web, web pages are associated to a social context that can tell us about their content, e.g. annotations, comments, etc.
2. "Common collaborative vocabularies" are needed to support common understanding since for a document; since each user has his own understanding of its content.
3. "Relevance relativeness" is needed since relevance is actually relative for each user.

The approach we are proposing relies on social annotations as source of social information, which are associated to documents in bookmarking systems. These latter are based on the techniques of *social tagging*. The principle is to provide the user with a mean to freely annotate resources on the Web with tags, e.g. URIs in *delicious*, or images in *flickr*. These annotations can be shared with others. This unstructured approach to classification is often referred to as a *folksonomy*.

The intuition behind the approach is that textual content of a document is expected to be a shared and a common representation for all users, while the social annotations used by a given user represent his own and personal understanding of its content, i.e. personal representation of this document. In this paper, our objective can then be formulated by the following question: *How to formalize a personal representation of a document in a social collaborative setting to improve web search?* Our contributions are the following:

1. A document representation called Personalized Social Document Representation (PSDR).
2. A ranking function of documents using their PSDRs w.r.t a query issued by a given user.
3. An evaluation study of our approach and a comparison with the closest works on a large public dataset.

The rest of this paper is organized as follows: in Section 2 we introduce the PSDR approach and the way it is used for ranking documents w.r.t a query. Section 3 presents the different experiments for evaluating our approach against the closest state of the art approaches. Finally, we conclude and provide some future directions in Section 4.

*This work has been mainly done when the authors was at Bell Labs France, Centre de Villarceaux.

[1]In this paper, we also refer to documents as web pages or resources.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

2. PSDR APPROACH

To illustrate our approach, let's consider a user *Bob,* who issues a query for which a number of web pages are retrieved, e.g. *YouTube.com*. Our approach intends to create a PSDR for each of these web pages according to *Bob* based on social annotations. For a given web page, the only consideration of the user's tags as his personalized representation will result either in: (i) ignoring this web page if he did not annotate it (a user doesn't tag all web pages) or (ii) assigning it an inappropriate ranking score (since the representation is only based on his own perspective, it may be poor). Our goal is then to use other users' annotations to enrich the personalized representation of the query issuer enabling him to: (i) benefit from others' experiences and feedbacks, (ii) promote used/visited resources even if they are not well classified, and (iii) discover new resources. Our approach proceeds into three main phases as illustrated in Figure 1:

1. Representing each document that matches the query using a Users-Tags matrix. This matrix is first sized, then weighted, in four steps enumerated from 1 to 4.
2. Using a matrix factorization process to infer the PSDR of a document that match the query to the query issuer based on the identification of weighting patterns. This phase is illustrated in the step 5 of Figure 1.
3. Finally, ranking documents based on their PDSR. This phase is illustrated in steps 6 and 7 of Figure 1.

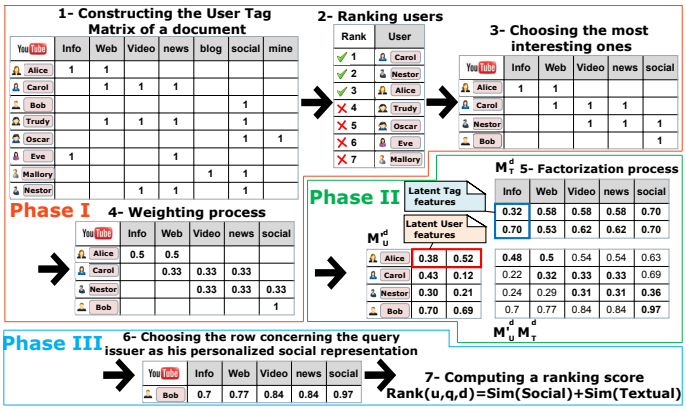


Figure 1: Process of creating a PSDR of a web page.

2.1 Constructing the Users-Tags matrix

2.1.1 *Sizing the Users-Tags matrix*

The objective in this first step is to gather as much useful information as possible around the user and the relatives who may serve to construct and enrich the PSDR. As illustrated in Figure 1, each web page can be represented using an $m \times n$ Users-Tags matrix $M^d_{U,T}$ of m users who annotate the web page and the n tags that they used to annotate it. Each entry w_{ij} represents the number of times the user u_i used the term t_j to annotate the considered web page. Note that a stemming is performed over terms before building the Users-Tags matrix.

Instead of using all users' feedback to infer a PSDR of the considered web page to *Bob*, we propose to choose only the most representative ones. Therefore, we use a ranking function to rank users from the most relevant to the less relevant ones, and select the top users as the most representative to both the query issuer and the considered web page (see Step 2 of Figure 1). This is to filter out irrelevant users who may represent noise. The ranking score of a user u according to a document d and the query issuer u_q is computed as:

$$Rank^d_{u_q}(u) = \overbrace{\alpha \times log\left(\frac{\mid D \mid}{\mid D_u \mid}\right) \times \frac{\mid T_{u,d} \mid}{\mid T_d \mid}}^{\text{Proximity to the document}} + \underbrace{(1-\alpha) \times cos(u, u_q)}_{\text{Proximity to the query issuer}} \quad (1)$$

where $\mid D \mid$ and $\mid D_u \mid$ are respectively the number of documents, and the number of documents tagged by u, $\mid T_d \mid$ and $\mid T_{u,d} \mid$ are respectively the number of tags of d, and the number of tags used by u to annotate d, $Cos(u, u_q)$ denotes the cosine similarity between a user who annotates d and the query issuer based on the tags they used, and α is a weight set to 0.5.

Once we get a ranked list of users, we select the top k as the most representative ones to both the considered document and the query issuer. Then, we select their tags to build a new (smaller) Users-Tags matrix $M^d_{U,T}$. Finally, we add the query issuer as a new entry in the Users-Tags matrix $M^d_{U,T}$ as well as his tags, if any (see step 3 of Figure 1).

2.1.2 *Weighting the Users-Tags matrix*

Our approach relies on its ability to compute for a given document d, an $m \times n$ Users-Tags matrix of m users and n tags where w_{ij} represents the extent to which the user u_i believes that the term t_j is associated to the document d. The main challenge here is *how to effectively estimate the personal weight of a tag t_j in a document d according to a user u_i?* We propose to use an adaptation of the well-known *tf-idf* measure to estimate this weight as follows:

$$w_{ij} = \frac{n^d_{u_i,t_j}}{\mid T_{u_i,d} \mid} \times log\left(\frac{|D_{u_i}|+1}{|D_{u_i,t_i}|}\right) \quad (2)$$

where $n^d_{u_i,t_j}$ is the number of times u_i used t_j to annotate d (computed after stemming), and $\mid D_{u_i,t_i} \mid$ is the number of documents tagged by u using t_i.

At the end of this step, we obtain a relatively smaller matrix capturing the closest users (and their tags) to the query issuer for each document that matches the query. Intuitively, the query issuer may have never annotated one of these documents, since the distribution of web pages over users follow a power law in folksonomies [14]. Given that, and due to the fact that a user is in average expected to use few terms to annotate a web page, we propose to infer a PSDR of a web page to a user based on other user's feedback, translated by the inference of missing values in the Users-Tags matrix. This inference process is operated through matrix factorization as detailed in the next section.

2.2 Computing PSDRs

Matrix factorization has proven its effectiveness in both quality prediction and scalability to predict missing values in sparse matrices [9, 10, 11]. This technique is based on the reuse of other users experience. Hence, to predict missing values in the Users-Tags matrix, it is first factorized into two latent matrices of features of users and tags by identifying weighting patterns. These latent features matrices are then

used to make further missing values prediction. Therefore, the Users-Tags matrix $M^d_{U,T}$ of a web page is factorized using $M^{'d}_U \times M^d_T$, where the matrix M^d_U denotes the user latent features, and M^d_T represents the tag latent features.

As an example, if we use 2 dimensions to factorize the matrix obtained in Step 4 of Figure 1 for weighting prediction, we obtain the matrices illustrated in Step 5 of Figure 1. Note that $M^d_{u_i}$ and $M^d_{t_j}$ are the column vectors and denote the latent feature vectors of user u_i and tag t_j for the web page *Youtube.com*, respectively. Then we can predict missing values w_{ij} using $M^{'d}_{u_i} M^d_{t_j}$. Each row i of the predicted matrix represents the PSDR of the i^{th} user of this web page. Notice that even if a user doesn't annotate a web page, this approach still can predict reasonable weightings.

A matrix factorization seeks to approximate the Users-Tags matrix $M^d_{U,T}$ constructed above by a multiplication of l-rank factors, minimizing the sum-of-squared-errors objective function over the observed entries as follows:

$$\min_{M^d_U, M^d_T} \mathcal{L} = \min_{M^d_U, M^d_T} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (M^d_{u_i,t_j} - M^{'d}_{u_i} \times M^d_{t_j})^2 \quad (3)$$

where $M^d_U \in R^{l \times m}$ and $M^d_T \in R^{l \times n}$, I_{ij} is is equal to 1 if u_i used t_j to annotate the d_i and equal to 0 otherwise.

The optimization problem in Equation 3 minimizes the sum-of-squared-errors between observed and predicted weightings. A gradient-based algorithm can be easily applied to minimize this function. Once $M^d_{U,T}$ factorized, we can predict missing values using $M^{'d}_U \times M^d_T$. Then, we consider:

PROPOSITION 1*: The row that corresponds to the query issuer in the predicted matrix $M^{'d}_U \times M^d_T$ corresponds to his PSDR of the considered document.*

Storing the PSDR of each document for each user is not suitable in fact. This is like creating and storing an index structure for each user, which is disk space consuming. Therefore, we propose to execute this factorization process on the fly, i.e. at query time. The complexity analysis, performed in Section 2.4, shows that this approach scales linearly with the number of documents that match the query.

2.3 Ranking documents using PSDR

In this paper, we consider a retrieval process in which we retrieve documents whose textual content includes all the query terms. The *Apache Lucene* search engine currently handles this process in our implementation. Hence, with respect to this process, we propose to compute a ranking score of a document d that potentially match the terms of a query q issued by a user u as follows:

$$Rank(u, q, d) = \gamma \times Sim(\overrightarrow{q}, \overrightarrow{S_{d,u}}) + (1 - \gamma) \times SES(\overrightarrow{q}, \overrightarrow{d}) \quad (4)$$

where, γ is a weight that satisfies $0 \leq \gamma \leq 1$, $SES(\overrightarrow{q}, \overrightarrow{d})$ is the Search Engine Score that express the similarity between the document d and the query q (computed by *Lucene*), $\overrightarrow{S_{d,u}}$ is the PSDR of the document d according to the user u.

Inspired by the Vectorial Model, queries, documents and their PSDRs are modeled as vectors. Hence, similarities between these vectors are computed using the cosine measure. At the end of this process we obtain a list of ranked documents according to (i) a matching between the textual content of documents and the query, and (ii) the social interest of the user extracted from close relatives in the folksonomy.

2.4 Complexity analysis

As pointed in [10], since the distribution of tags and users over documents in folksonomies follows a power law, the Users-Tags matrix is expected to be extremely sparse. Hence, the total computational complexity in one iteration of the gradient descent algorithm is $O(\rho)$, where ρ is the number of nonzero entries in the Users-Tags matrix. Consequently, for factorizing one document, the computational complexity is estimated to be $O(i \times \rho)$, where i is the number of iteration of the gradient algorithm ($i \backsimeq 10$). Finally, the computational complexity for evaluating a query that match m documents is estimated to $O(m \times i \times \rho)$. Since i and ρ are estimated to very low values, we can say that the complexity scale linearly with the number of retrieved documents. By using parallel computation, we can easily and considerably reduce the execution time. This is part of our future work.

3. EVALUATION

In this section, we describe the dataset we used, the evaluation methodology and the evaluations we have performed.

3.1 Dataset

To evaluate our approach, we have selected a public *delicious* dataset, which is described and analyzed in [14]. Before the experiments, we performed four data preprocessing tasks: (1) We remove annotations that are too personal or meaningless, e.g. "toread", "Imported IE Fa-vorites", etc. (2) The list of terms undergoes a stemming by means of the Porter's algorithm in such a way to eliminate the differences between terms having the same root. (3) We downloaded all the available web pages while removing those, which are no longer available using the *cURL* command line tool. (4) Finally, we removed all the non-English web pages. Table 1 gives a description of the resulted dataset:

Table 1: Details of the delicious dataset

Bookmarks	Users	Tags	Web pages	Unique terms
9 675 294	318 769	425 183	1 321 039	12 015 123

3.2 Evaluation methodology

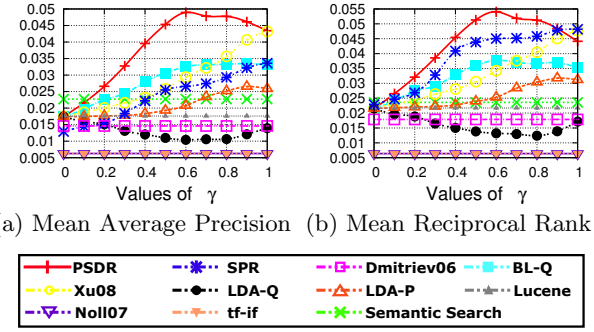
Making evaluations for personalized search is a challenge since relevance judgements can only be assessed by end-users themselves, which is difficult to achieve at a large scale. However, different efforts [3, 4] state that the tagging behavior of a user of folksonomies closely reflects his behavior of search on the Web. In other words, if a user tags a document d with a tag t, he will choose to access the document d if it appears in the result obtained by submitting t as query to a search engine. Thus, we can easily state that any triple (u, t, d) that represents a user u who tagged a document d with tag t, can be used as a test query for evaluations. The main idea of these experiments is based on the following assumption: *For a query $q = \{t\}$ issued by user u with term t, relevant documents are those tagged by u with t.*

Hence, for each evaluation, we randomly select 2000 pairs (u, t), which are considered to form a personalized query set. For each corresponding pair (u, t), we remove all the triplets (u, t, d) in order to not promote the document d in the results obtained by submitting t as a query in our algorithm and the considered baselines. For each pair, the user u sends the query $q = \{t\}$ to the system. Then, we

retrieve and rank all the documents that match this query using our approach or a specific baseline, where documents are indexed using the Apache Lucene. Finally, according to the previous assumption, we compute the Mean Average Precision (MAP) and the Mean Reciprocal Rank (MRR) over the 2000 queries. The random selection was carried out 10 times independently, and we report the average results.

3.3 Comparison with baselines

In a previous evaluation that we performed for studying the impact of the top k closest users, we conclude that optimal performance is obtained while selecting two users for building the Users-Tags matrix. Thus, our approach is evaluated using the most two related users, and 5 dimensions for the factorization process. We compare our approach to several personalized and non-personalized baselines, in which the social based score is merged with the textual based matching score using a linear function with a γ parameter. The results are illustrated in Figure 2, while varying γ.



(a) Mean Average Precision (b) Mean Reciprocal Rank

Figure 2: Comparison with the baselines.

3.3.1 PSDR VS non-personalized approaches

We compare our approach to: SocialPageRank (SPR) [1], Dmitriev06 [8], and the Lucene naive score. We also compare our approach to an approach where the matching score is computed as in Equation 4, but the social representation of documents is based on all annotations weighted using the *tf-idf* measure (BLQ). The last approach use LDA [5], where we model queries and documents. Then, for each document that match a query, we compute a similarity between its topic and the topic of the query using the cosine measure. The obtained value is merged with the textual ranking score as in Equation 4 (LDA-Q).

The results show that our approach is much more efficient than all the non-personalized approaches for all values of γ. Hence, we conclude that the personalization efforts introduced by our approach in the representation of documents with respect to each user bring a considerable improvement of the search quality. We also notice that most of the non-personalized approaches decease their performance for high values of γ. This is certainly due to the fact that they are not designed for personalized search, since these approaches fail in discriminating between users in spite of their preferences.

3.3.2 PSDR VS personalized approaches

Here we compare our approach to: Xu08 [15], Noll07 [12], tf-if [13], and Semantic Search [2]. We also propose an approach based on LDA to model users and documents, in which for each document that match a query, we compute a similarity between its topic and the topic of the user profile using the cosine measure. The obtained value is merged with the textual ranking score (LDA-P). The obtained results also show that our approach is much more efficient than all the baselines for all values of γ. Especially, our approach outperform the LDA-P approach and the Xu08 approach, which we consider as the closest works to our. We also notice that the Noll07 and the tf-if approaches give poor results. This is certainly due to the fact that they fail in ranking documents that doesn't share tags with the query issuer.

4. CONCLUSION AND FUTURE WORK

This paper discusses a contribution to the area of IR modeling while leveraging the social dimension of the web. We proposed a Personalized Social Document Representation approach, an attempt to use social information to enhance and improve documents for users. When a user submits a query, we construct, on the fly, a PDSR of all documents that potentially match the query based on other users experience. Then, we rank these documents with respect to the computed PSDR. The experiments that we have performed on a *delicious* dataset show the benefit of such an approach.

Currently, we are investigating ways to add social regularization terms to the objective function to constrain it and reduce the solution space. The temporal dimension of social users' behavior has not been investigated yet and is part of our future work. Finally, performing an online user evaluation to validate our results is on-going. PSDR has been developed and integrated to the LAICOS [6] platform.

5. REFERENCES

[1] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW*, 2007.

[2] M. Bender, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, R. Schenkel, and G. Weikum. Exploiting social relations for query expansion and result ranking. In *ICDE Workshops*, 2008.

[3] D. Benz, A. Hotho, R. Jäschke, B. Krause, and G. Stumme. Query logs as folksonomies. *Datenbank-Spektrum*, 2010.

[4] K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. Can all tags be used for search? In *CIKM*, 2008.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[6] M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub. LAICOS: An Open Source Platform for Personalized Social Web Search. In *KDD*, 2013.

[7] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2), September 2002.

[8] P. A. Dmitriev, N. Eiron, M. Fontoura, and E. Shekita. Using annotations in enterprise search. In *WWW*, 2006.

[9] D. Dueck, B. J. Frey, D. Dueck, and B. J. Frey. Probabilistic sparse matrix factorization. Technical report, 2004.

[10] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, 2008.

[11] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, 2011.

[12] M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. In *ISWC'07/ASWC'07*, 2007.

[13] D. Vallet, I. Cantador, and J. M. Jose. Personalizing web search with folksonomy-based user and document profiles. In *ECIR*, 2010.

[14] R. Wetzker, C. Zimmermann, and C. Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *ECAI*, 2008.

[15] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In *SIGIR*, 2008.