PerSaDoR: Personalized Social Document Representation for Improving Web Search

Mohamed Reda Bouadjenek[a,*], Hakim Hacid[b,*], Mokrane Bouzeghoub[c], Athena Vakali[d]

[a]*Department of Computing and Information Systems, University of Melbourne, Victoria, Australia.*
[b]*Zayed University, United Arab Emirates.*
[c]*DAVID Laboratory, University of Versailles-Saint-Quentin-en-Yvelines (UVSQ), France.*
[d]*Computer Science Department, Aristotle University of Thessaloniki, Greece.*

Abstract

In this paper, we discuss a contribution towards the integration of social information in the index structure of an IR system. Since each user has his/her own understanding and point of view of a given document, we propose an approach in which the index model provides a Personalized Social Document Representation (PerSaDoR) of each document per user based on his/her activities in a social tagging system. The proposed approach relies on matrix factorization to compute the PerSaDoR of documents that match a query, at query time. The complexity analysis shows that our approach scales linearly with the number of documents that match the query, and thus, it can scale to very large datasets. PerSaDoR has been also intensively evaluated by an offline study and by a user survey operated on a large public dataset from *delicious* showing significant benefits for personalized search compared to state of the art methods.

Keywords: Information Retrieval, Social Networks, Social Information Retrieval, Social Search, Social Recommendation.

1. Introduction

With the fast growing of the social Web, users are becoming more active in generating content through blogging and content characterization on social platforms like *Facebook*[1] and *Twitter*[2] using comments, tags, ratings, shares, etc. A crucial problem is then to enable users to find relevant information with respect to their interests and needs. Information Retrieval (IR) is performed every day in an obvious way over the Web, typically using a search engine. However, finding relevant information remains challenging for end-users as: (i) usually, a user doesn't necessarily know what he/she is looking for until he/she finds it, and (ii) even if a user knows what he/she is looking for, he/she does not always know how to formulate the right query to find it (except in the case of navigational queries [11]). In existing IR systems, queries are usually interpreted

*This work has been primarily completed while the authors were at Bell Labs France, Centre de Villarceaux, 91620 Nozay.
Email addresses: reda.bouadjenek@unimelb.edu.au (Mohamed Reda Bouadjenek), hakim.hacid@zu.ac.ae (Hakim Hacid), mokrane.bouzeghoub@uvsq.fr (Mokrane Bouzeghoub), avakali@csd.auth.gr (Athena Vakali)

[1] https://www.facebook.com/

[2] https://twitter.com/

and processed using indexes and/or ontologies, which are hidden to users. The resulting documents[3] are not necessarily relevant from an end-user perspective, in spite of the ranking performed by the Web search engine.

To improve the IR process and reduce the amount of irrelevant documents, there are mainly three possible improvement tracks: (i) query reformulation using extra knowledge, i.e., expansion or refinement of a query, (ii) post filtering or re-ranking of the retrieved documents (based on the user profile or the context), and (iii) improvement of the IR model, i.e., the way documents and queries are represented and matched to quantify their similarities. This third track is the focus of this work since it has been the least explored in the recent literature. We will focus in particular on enhancing the representation of documents for personalized search. This is achieved by considering social metadata related to documents and users on social tagging systems. We provide in the following the motivation behind our focus on the third mentioned track.

1.1. Motivation

Our motivations to improve the IR model are mainly driven by the following observations:

1. A "social contextual summarization" is required as Web pages are associated to a social context that can tell a lot about their content (e.g., social annotations). Several studies have reported that adding a tag to the content of a document enhances the search quality, as they are good summaries of documents [4, 12, 16, 50] (e.g., document expansion [21]). In particular, social information can be useful for documents that contain few terms where a simple indexing strategy is not expected to provide a good retrieval performance (e.g., the *Google* homepage[4]).
2. "Common collaborative vocabularies" are needed to support a common understanding since for a given document, each user has his/her own understanding of its content. Therefore, each user uses a different vocabulary and different words to describe, comment, and annotate this document. For example, if we consider the *YouTube* homepage[5], a given user can tag it using "video", "Web" and "music" while another user can tag it using "news", "movie", and "media".
3. "Relevance relativeness" is needed since relevance is actually specific to each user. Hence, adapting search results according to each user in the ranking process is expected to provide good retrieval performance.

Following by these observations, we believe that enhancing the representation of documents and personalizing them with social information is expected to improve Web search. Exploiting social information has also a number of advantages (for IR in particular): First, feedback information in social networks is provided

[3] We also refer to documents as Web pages or resources.

[4] `http://www.google.com/`

There are only a very few terms on the page itself but a thousands of annotations available on *delicious* are associated to it. Eventually, the social information of the *Google* homepage is more useful for indexing.

[5] `http://www.youtube.com/`

directly by the user, so accurate information about users' interests can be harvested as people actively express their opinions on social platforms. Second, a huge amount of social information is published and available with the agreement of the publishers. Exploiting this information should not violate user privacy particularly when referring to social tagging information, which does not contain sensitive information about users. Finally, social resources are often publicly accessible, as most of social networks provide APIs to access their data (even if often a contract must be established before any use).

1.2. Problem definition and contributions

Our approach in this work is an extension of the basic one proposed in [10]. We rely on users' annotations as a source of social information, which are associated to documents in bookmarking systems. As illustrated in Figure 1, the textual content of a document is shared between users under a common representation, i.e., all terms in a document are identically shared and presented to users as in the classic Vector Space Model (VSM), while the annotations given by a user to this document express his/her personal understanding of its content. Thus, these annotations express a personal representation of this document to this user. For example, as illustrated in Figure 1, the red annotations given by *Bob* to the document express his personal representation/view of this document. On the other hand, the green annotations constitute the personal representation of this document to Alice since she has used them to describe the document's content. In this paper, our main objective is to answer the following question: *How to formalize a personal representation of a document in a social collaborative setting, and how to use this representation in document search to, hopefully, improve the search quality?*

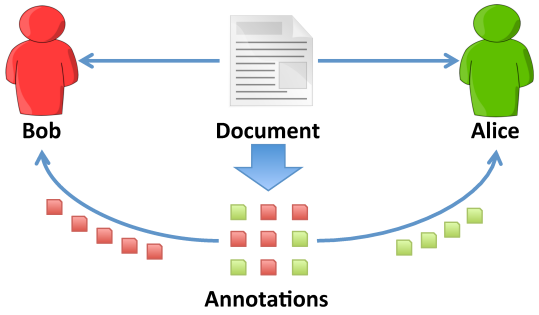


Figure 1: Document representations for two users.

The problem we are addressing in this paper is strongly related to personalization since we want to: (i) formalize personal representations of documents, and (ii) propose adapted search results. Personalization allows differentiating between users by emphasizing on their specific domains of interest and their preferences. Personalization is a key element in IR and its demand is constantly increasing by numerous users for adapting their search results. Several techniques exist to provide personalized services among which the user profiling. The user profile is a collection of personal information associated to a specific user that enables to capture his/her interests. Details of how we model user profiles are given in Section 3.

In this perspective, we propose the following contributions:

1. A document representation called Personalized Social Document Representation (PerSaDoR) which is based on social information that is collected in social bookmarking systems. The PerSaDoR is expected to deliver, for a given document, different social representations according to each user based on the feedback of other users.
2. A key problem in an IR model is the definition of a ranking function used to establish a simple ordering of the documents retrieved. Hence, we propose two ranking functions that take into account both the textual content of documents and their PerSaDoR according to the query issuer.
3. Our approach is validated by an offline study and a user survey on a large public dataset. This shows to which extent our approach contributes to an efficient Web search at the expense of existing approaches. The complexity analysis shows that our approach can be applied to large datasets since it scales linearly with the number of documents that match the query.

1.3. Paper organization

The rest of this paper is organized as follows: in Section 2 we present the related work. Next, in Section 3 we present the concepts and the notation used throughout this paper. Section 4 introduces our approach of Personalized Social Document Representation and our ranking functions. In Sections 5, 6, and 7, we discuss the different experiments that evaluate the performance of our approach. Finally, we conclude and provide some future directions in Section 8.

2. Related Work

The current models of information retrieval are blind to the social context that surrounds Web pages and users. Therefore, recently, the fields of Information Retrieval (IR) and Social Networks Analysis (SNA) have been bridged resulting in *s*ocial information retrieval (SIR) models. These models are expected to extend conventional IR models to incorporate social information [8]. In this paper, we are mainly interested in how to use social information to improve classic Web search, and in particular the representation of documents and the re-ranking of documents. Hence, we review in the following the research work related to these two aspects.

2.1. Indexing and Modeling Using Social Information

Throughout our analysis of the state of the art , we have noticed that social information has been mainly used in two ways for modeling and enhancing documents' representations: (i) either by adding social metadata to the content of documents, e.g., document expansion, or (ii) by personalizing the representation of documents, following the intuition that each user has his/her own vision of a given document.

2.1.1. Document Expansion (Non-Personalized Indexing)

In [12, 16, 15], authors propose to index a document with both its textual content and its associated tags modeled as in the VSM. However, each method uses a different algorithm for weighting social metadata, e.g., tf-idf [15], term quality [12], etc. Also, Zhang et al. [50] proposed a framework to enhance documents' representations using social annotations. The framework consists in representing a document in a dual-vector representation: (i) enhanced textual content vector and (ii) enhanced social content vector; each component being calculated from the other. A more recent work by Nguyen et al. [33] proposed a framework named SoRTESum to combine Web document contents, sentences and users' comments from social networks to provide a viewpoint of a Web document towards a special event. SoRTESum obtained improvements over state of the art supervised and unsupervised baselines to generate high-quality summaries. An interesting future work is to use the obtained summaries for querying the documents.

2.1.2. Personalized Indexing and Modeling of Documents

Amer-Yahia et al. [1] investigated efficient Top-k processing in collaborative tagging sites. The idea is that the score of an answer is estimated by its popularity among the members of a seeker's network. Basically, the solution is to create personalized indexes based on clustering strategies, which achieve different compromises between storage space and processing time. In the same spirit, Servajean et al. [36] proposed a simplified profile diversification model and different diversification algorithms have been used to compute the score of an item during the processing of a query. The proposed approach reduces significantly the number of accesses to the inverted lists done by the Top-k algorithm.

Finally, Xu et al. [43] proposed a dual personalized ranking function which adopts two profiles: an extended user profile and a personalized document profile. Briefly, for each document the method computes for each individual user a personalized document profile to better summarize his/her perception about it. The proposed solution estimates this profile based on the perception similarities between users.

2.2. Document re-ranking

We can distinguish two categories for social results re-ranking that differ in the way they use social information. The first category uses social information by adding a social relevance to documents while the second uses it for personalization.

2.2.1. Non-Personalized Ranking

Social relevance refers to information socially created that characterizes a document from an interest point of view, i.e., its general interest, its popularity, etc. Two formal models for folksonomies and ranking algorithm called *folkRank* [22] and *SocialPageRank* [2] have been proposed. Both are an extension of the well-known *PageRank* algorithm adapted for the generation of rankings of entities within folksonomies. *SocialPageRank* intends to compute the importance of documents according to a mutual enhancement relation among popular resources, up-to-date users, and hot social annotations. In the same spirit, relying

on social bookmarking systems, Takahashi et al. [39] proposed *S-BIT* and *FS-BIT*, which are extensions of the well-known HITS algorithm [23]. Yanbe et al. [44] proposed *SBRank* which indicates how many users bookmarked a page, and use the estimation of *SBRank* as an indicator of Web search.

The work in [17] proposed a method to use microblogging data stream to compute novel and effective features for ranking fresh URLs, i.e., "uncrawled" documents likely to be relevant to queries where the user expects documents which are both topically relevant as well as fresh. The proposed method consists of a machine-learning based approach that predicts effective rankings for query-url pairs. Recently He et al. [20] proposed a new method to predict the popularity of items (i.e., Webpages) based on users' comments, and to incorporate this popularity into a ranking function. Yang et al. [45] proposed SESAME, a fine-grained preference-aware social media search framework leveraging user digital footprints on social networks. The proposed method is based on users' direct feedback obtained from their social networks, their sentiment about the media content, and the associated keywords from their comments to characterize their fine-grained preference. Then, they use a parallel multi-tuple based ranking tensor factorization algorithm to perform a personalized media item ranking. The results show that SESAME can subtly capture user preferences on social media items and consistently outperform baseline approaches by achieving better personalized ranking quality.

In the context of graph mining, Siersdorfer et al. [38] introduced novel methodologies for query based search engine mining which enable efficient extraction of social networks from large amounts of Web data. To this end, they used patterns in phrase queries for retrieving entity connections, and employed a bootstrapping approach for iteratively expanding the pattern set. The experimental evaluation in different domains demonstrates that the proposed algorithms provide high quality results and allow for scalable and efficient construction of social graphs.

In the context of image search, image search engines like Google and Bing usually adopt textual information to index images. Although the performance is acceptable for many queries, the accuracy of retrieved images is still not high in most cases. The probable mismatch between the content of an image and the text from a web page is a major problem. Indeed, the extracted text does not always precisely describe the characteristics of the image content, as required by the query. Interesting solutions proposed by Yu et al. [47, 46, 48] to address this problem aim to integrate visual information of images into a learning to rank framework. Also, the work by Lai et al. [26] proposes to learn the ranking model which is constrained to be with only a few nonzero coefficients using ℓ_1-regularization constraint and propose a learning algorithm from the primal dual perspective. A more recent work by Zhang et al. [49] optimized the max-margin loss on triplet units to learn deep hashing function for image retrieval.

2.2.2. Personalized Ranking

Several approaches have been proposed to personalize ranking of search results using social information [3, 7, 13, 34, 35, 37, 40, 42]. Almost all these approaches are in the context of folksonomies and follow a common idea that the ranking score of a document d retrieved when a user u submits a query Q is driven

by: (i) a term matching, which calculates the similarity between Q and the textual content of d to generate a user unrelated ranking score; and (ii) an interest matching, which calculates the similarity between u and d to generate a user related ranking score. Then a merge operation is performed to generate a final ranking score based on the two previous ranking scores.

Kumar et al. [25] proposed two methods to build a Clustered User Interest Profile for each user, using a set of tags. A profile contains many clusters, and each cluster identifies a topic of the user's interest. The matching cluster associated with the given user's query, aids in the disambiguation of user search needs and assists the search engine to generate a set of personalized search results. Finally, a more recent work by Du et al. [18] proposed a new multi-level user profiling model by integrating tags and ratings to achieve personalized search, which can reflect not only a user's likes but also a his/her dislikes. The obtained results showed significant improvement for MRR compared to several baseline methods.

3. Background and notations

In this section, we formally define the basic concepts that we use throughout this paper, namely, a bookmark, a folksonomy, and a user profile. These concepts are crucial in the problem of IR modeling in social bookmarking systems and their formulation will support the proposed work contributions.

Social bookmarking websites are based on the techniques of *social tagging* and *collaborative tagging*. The principle behind social bookmarking platforms is to provide the user with a mean to annotate resources on the Web, e.g., URIs in *delicious*, videos in *YouTube*, images in *Flickr*, or academic papers in *CiteULike*. These annotations (also called tags) can be shared with others. This unstructured (or better, free structured) approach to classification with users assigning their own labels is often referred to as a *folksonomy* [19]. A folksonomy is based on the notion of bookmark which is formally defined as follows:

Definition 1. [Bookmark] Let U, T, R be respectively the sets of Users, Tags, and Resources. A *bookmark* is a triplet (u, t, r) such as $u \in U, t \in T, r \in R$ which represents the fact that the user u has annotated the resource r with the tag t.

Then, a folksonomy is formally defined as follows:

Definition 2. [Folksonomy] Let U, T, R be respectively the sets of Users, Tags and Resources. A folksonomy $\mathbb{F}(U, T, R)$ is a subset of the cartesian product $U \times T \times R$ such that each triple $(u, t, r) \in \mathbb{F}$ is a *bookmark*.

A folksonomy can then be naturally represented by a tripartite-graph where each ternary edge represents a bookmark. In particular, the graph representation of the folksonomy $\mathbb{F}$ is defined as a tripartite graph $\mathcal{G}(V, E)$ where $V = U \cup T \cup R$ and $E = \{(u, t, r) | (u, t, r) \in \mathbb{F}\}$. Figure 2 shows nineteen bookmarks provided by eight users on one resource using seven tags.

Folksonomies have proven to be a valuable knowledge for user profiling [9, 13, 34, 40, 42]. Especially, because users tag interesting and relevant information to them with keywords that may constitute a good

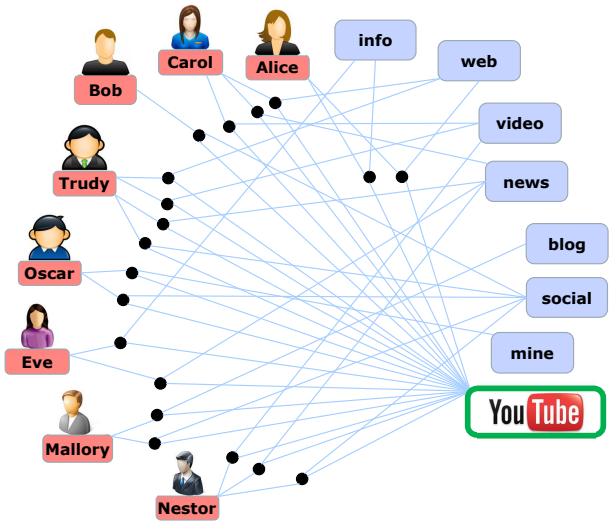


Figure 2: Example of a folksonomy with eight users who annotate one resource using seven tags. The triples (u, t, r) are represented as ternary-edges connecting a user, a resource and a tag.

summary of their interests. Hence, in this paper and in the context of folksonomies, a profile includes all the terms used as tags along with their weights to capture user's tagging activities. It is defined as follows:

Definition 3. [User Profile] Let U, T, R be respectively the set of Users, Tags and Resources of a folksonomy $\mathbb{F}(U, T, R)$. A profile assigned to a user $u \in U$, is modeled as a weighted vector $\overrightarrow{p_u}$ of m dimensions, where each dimension represents a tag the user employed in his/her tagging actions. More formally, $\overrightarrow{p_u} = \{w_{t_1}, w_{t_2}, ..., w_{t_m}\}$ such that $t_m \in T \wedge (\exists r \in R \mid (u, t_m, r) \in \mathbb{F})$, and w_{t_m} is the weight of t_m computed using an adaptation of the well-known *tf-idf* measure as in [9].

Finally, throughout this paper we use the notations summarized in Table 1.

4. PerSaDoR: Personalized Social Document Representation

In this section, we first give an insight of our approach using a simple toy example. Then, we introduce our PerSaDoR method. Finally, we show how to use a PerSaDoR for ranking documents.

4.1. Toy example and approach overview

Before going into the details of our approach, we describe hereafter a scenario to illustrate our proposal throughout this paper.

Example 1. Suppose that a user, say *Bob*, issues the query *"news on the Web"* for which a number of Web pages are retrieved. Let's consider the Web page *YouTube.com* as a document that matches this query. This Web page is associated with many bookmarks in a folksonomy as illustrated in Figure 2. There are eight users (*Alice, Bob, Carol, Eve, Mallory, Nestor, Oscar,* and *Trudy*) who annotated *YouTube.com* using seven tags (*info, Web, video, news, blog, social,* and *mine*).

Table 1: Summary of the Paper's Notation.

Variable	Description	Variable	Description
u, d, t	Respectively a user u, a document d, and a tag t.	U_t, U_d, $U_{t,d}$	Respectively the set of users that use t, users that annotate d, and users that used t to annotate d.
U, D, T	Respectively a set of users, documents, and tags.	$M^d_{U,T}$	The Users-Tags matrix associated to the document d.
$\mid A \mid$	The number of element in the set A.	M^d_U, M^d_T	Respectively the user latent feature matrix, and the tag latent feature matrix associated to a document d.
T_u, T_d, $T_{u,d}$	Respectively the set of tags used by u, tags used to annotate d, and tags used by u to annotate d.	$\overrightarrow{p_u}$	The weighted vector of the profile of the user u.
D_u, D_t, $D_{u,t}$	Respectively the set of documents tagged by u, documents tagged with t, and documents tagged by u with t.	$\| . \|_F$	The Frobenius norm where: $\|M\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} \mid a_{ij} \mid^2}$

Our approach intends to create, on the fly, a representation for each of these retrieved Web pages from the perspective of *Bob* based on their associated social annotations. These representations are used in order to compute a ranking score w.r.t. the query. Since a given document representation is specific to *Bob*, it is by definition personalized and we call it from now on, a *Personalized Social Document Representation (*PerSaDoR*)*.

For a given Web page (e.g., *YouTube.com*), the only consideration of the user's tags as his/her personalized representation will result either in: (i) ignoring this Web page if he/she didn't annotate it or (ii) assigning it an inappropriate ranking score (since the representation is only based on his/her own perspective which 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 feedback, (ii) promote used/visited resources even if they are not well classified, and (iii) discover new resources.

For a document that potentially matches a query, our method proceeds into three main phases in order to collect maximum useful information about this document and its social relatives. This information is reused to create its PerSaDoR according to a query issuer. These phases are the following, as illustrated in Figure 3:

1. Representing each document that matches the query terms using a Users-Tags matrix. This matrix is first sized by selecting relevant users to the query issuer, e.g., *Carol, Nestor, and Alice*. Then, each entry of the Users-Tags matrix is computed by estimating the extent to which the user would associate

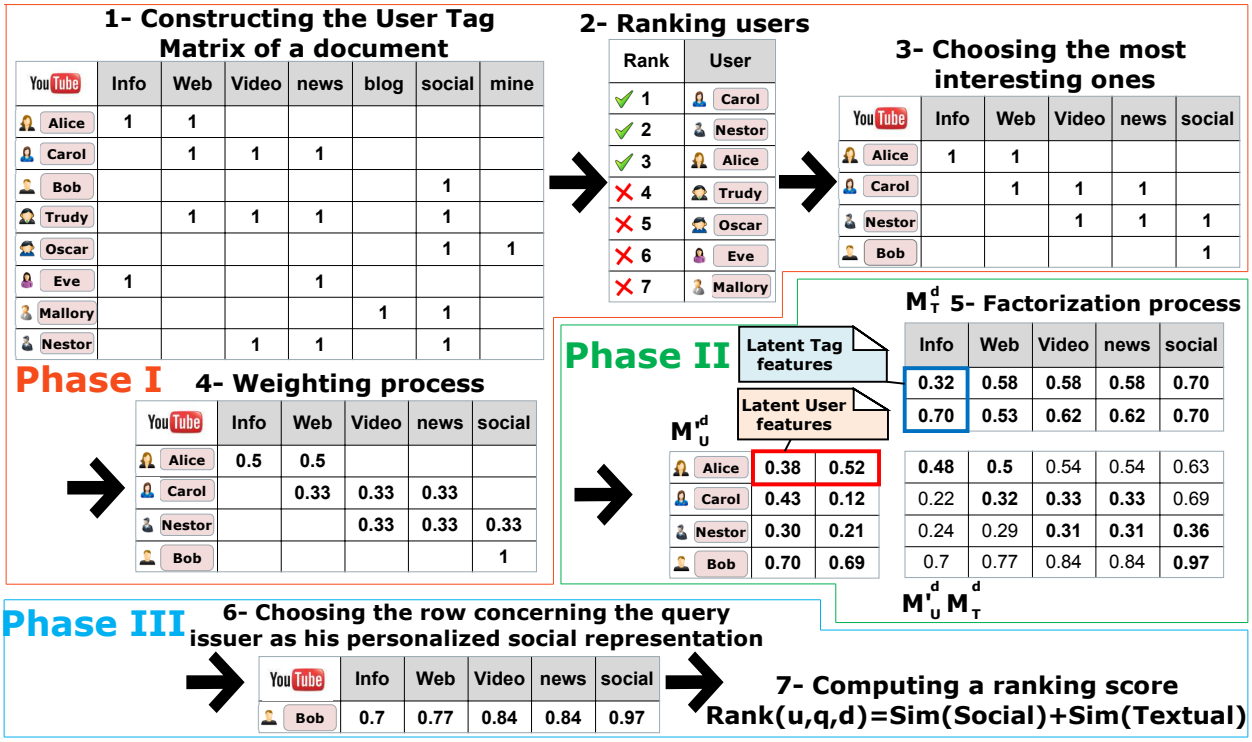


Figure 3: Process of creating a personalized social representation of the Web page *YouTube.com* to the user *Bob* of the folksonomy of Figure 2.

the tag to the considered document, e.g., *Alice* thinks that *info* is associated to *YouTube.com* with a weight of 0.5. This phase includes four sub-steps enumerated from 1 to 4 in Figure 3.

2. Each row i in a Users-Tags matrix of a given document translates the personal representation of the user u_i. This matrix is expected to be sparse, since it contains many missing values that should be inferred to build the PerSaDoR for the query issuer. Hence, a matrix factorization process is used to infer the PerSaDoR of the considered document to the query issuer based on identifying weighting patterns. This phase corresponds to step 5 in Figure 3.
3. Finally, ranking documents based on their PerSaDoR and their textual content. This phase is illustrated in steps 6 and 7 in Figure 3.

We detail in the following these different phases illustrated with our toy example.

4.2. Constructing the Users-Tags matrix

We detail here how we represent a Web page using a Users-Tags matrix, and how it is weighted. This matrix will be subsequently used to infer the PerSaDoR of the considered Web page w.r.t. a query issuer.

4.2.1. Sizing the Users-Tags matrix

The objective in this first step is to gather as much useful information as possible about the user and the social relatives who may serve to construct and enrich the PerSaDoR. As illustrated in Figure 3, 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} in the matrix represents the number of times the user u_i used the term t_j to annotate the considered Web page.

Example 2. In the folksonomy of Figure 2, *Bob* used the term *video* to annotate the Web page *YouTube.com* once. A stemming is performed over terms before building the User-Tag matrix. Hence, if a user uses the terms *new* and *news* to annotate a Web page, we consider only the term *new*, and we put the value 2 in the entry that corresponds to this user and this term when building the matrix.

Instead of using all users' feedback to infer a PerSaDoR of the considered Web page to *Bob*, we propose to select only the most representative ones in order to filter out irrelevant users who may introduce noise. To do so, we use a ranking function to rank users from the most relevant to the less relevant ones, and select only the Top k users as the most representative ones to both the query issuer and the considered Web page (see Step 2 of Figure 3). The irrelevant users may:

1. have annotated a lot of documents improperly;
2. have annotated the considered document with few terms;
3. not be socially close to the query issuer and thus don't share the same topics of interests.

Then, we select only the terms that the Top k users employed to annotate this Web page and build a new reduced Users-Tags matrix, which is expected to be more representative to both the query issuer and the considered Web page (see Step 3 in Figure 3). Note that even if the query issuer has annotated the considered Web page, we do not consider him/her in the ranking process since we want to rank users with respect to him/her.

The ranking score of a user u according to a document d and the query issuer u_q is computed as follows:

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

where $sim(u, u_q)$ denotes the similarity between a user who annotates d and the query issuer. α is a weight that satisfies $0 \leq \alpha \leq 1$, which allows giving more importance to either the document proximity part or to the query issuer proximity part. As described in [31], the similarity between two users can be computed using one of the measures mentioned in Table 2.

Once we get a ranked list of users using Equation 1, we select the Top k to be the most representative ones to both the considered document and the query issuer. Then, we select their tags to built 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/her tags, if any (see step 3 of Figure 3). Once the matrix is built, we proceed to the computation of the weights associated to each entry as detailed in the next section.

Table 2: Similarity measures summarization (i.e., $Sim(u, u_q)$).

Dice	$Dice(u, u_q) = 2 \times \frac{\lvert T_u \cap T_{u_q} \rvert}{\lvert T_u \rvert + \lvert T_{u_q} \rvert}$
Jaccard	$Jacard(u, u_q) = \frac{\lvert T_u \cap T_{u_q} \rvert}{\lvert T_u \cup T_{u_q} \rvert}$
Overlap	$Overlap(u, u_q) = \frac{\lvert T_u \cap T_{u_q} \rvert}{min(\lvert T_u \rvert, \lvert T_{u_q} \rvert)}$
Cosine	$Cos(u, u_q) = \frac{\overrightarrow{P_u} \bullet \overrightarrow{P_{u_q}}}{\lvert \overrightarrow{P_u} \rvert \times \lvert \overrightarrow{P_{u_q}} \rvert}$

4.2.2. 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 with the document d.

Example 3. The tagging actions of *Alice* regarding the Web page *YouTube.com* can be summarized as mixtures of two tags, *Info* and *Web*. Therefore, we can suppose that the distribution of these two tags in this Web page according to *Alice* is 50% for *Info* and 50% for *Web*. We refer to the distribution of a tag t_j in a document d according to a user u_i as: *the personal weight of* t_j *in* d *according to* u_i.

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. Therefore, we define the weight w_{t_i} of the term t_i in a document d according to a user u_i as the *user term frequency, inverse document frequency (utf-idf)*, which is computed as follows:

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

where $n^d_{u_i,t_j}$ is the number of times u_i used t_j to annotate d (computed after stemming). A high weight in *utf-idf* is reached by a high user term frequency and a low document frequency of the term in the whole set of documents tagged by the user; the weights hence tend to filter out terms commonly used by a user (see Step 4 of Figure 3).

At the end of this step, we obtain a matrix capturing the closest users (and their tags) to the query issuer, and this for each document that potentially match the query. Intuitively, the query issuer may have never annotated one of these documents, since the distribution of Web pages over users follows a power law in folksonomies [21] (see Figure 4). Given that, and due to the fact that a user is expected to use few terms to annotate a Web page, we propose to infer a PerSaDoR of this Web page to that user based on other users feedback. This is translated by the inference of missing values in the Users-Tags matrix using matrix factorization as detailed in the next section.

4.3. Matrix factorization

In the previous steps, we showed how we represent a document that matches a query using a Users-Tags matrix. This latter is expected to contain as much relevant information as possible for the query issuer

and the document by selecting relevant users and their tags. Each row i in the Users-Tags matrix of a given document constitutes the personal representation of the user u_i. However, this matrix is sparse, since it contains many missing values that should be inferred to compute the PerSaDoR of the query issuer in particular. Therefore, the problem at this point is to predict these missing values effectively and efficiently by employing other users feedback. One way to do so is to use matrix factorization.

Matrix factorization has proven its effectiveness in both quality and scalability to predict missing values in sparse matrices [14, 27, 29, 28, 30]. This technique is based on the reuse of other users experience and feedback in order to predict missing values in a matrix. Concretely, to predict these missing values, the Users-Tags matrix is first factorized into two latent features matrices of users and tags. These latent features matrices are then used to make further missing values prediction. In its basic form, matrix factorization characterizes both users and tags by vectors of factors inferred from identifying weighting patterns. Therefore, the Users-Tags matrix $M^d_{U,T}$ of the Web page *YouTube.com* is factorized using $M^{'d}_U \times M^d_T$, where the low-dimensional matrix M^d_U denotes the user latent features, and M^d_T represents the low-dimensional tag latent features.

Example 4. If we use two dimensions to factorize the matrix obtained in Step 4 of Figure 3, we obtain the matrices illustrated in Step 5 of Figure 3. 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 $M^{'d}_{u_i} \times M^d_{t_j}$. Each row i of the predicted matrix $M^{'d}_U \times M^d_T$ represents the personal representation of the i^{th} user according to this Web page.

Notice that even if a user doesn't annotate a Web page, this approach still can predict reasonable weights as shown in Section 6.2. Also, it is important to mention that the solution of M^d_U and M^d_T is not unique (it depends on several parameters, e.g., the number of latent dimensions or the initial values of the factorization).

A matrix factorization seeks to approximate the Users-Tags matrix $M^d_{U,T}$ by a multiplication of l-rank factors, as follows:

$$M^d_{U,T} \approx M^{'d}_U \times M^d_T \tag{3}$$

where $M^d_U \in R^{l \times m}$ and $M^d_T \in R^{l \times n}$. Therefore, we can approximate the Users-Tags matrix $M^d_{U,T}$ by minimizing the sum-of-squared-errors objective function over the observed entries as follows:

$$\arg\min_{M^d_U, M^d_T} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (M^d_{u_i,t_j} - M^{'d}_{u_i} \times M^d_{t_j})^2 \tag{4}$$

where I_{ij} is the indicator function that is equal to 1 if user u_i used the tag t_j to annotate the document d and equal to 0 otherwise. In order to avoid overfitting in the learning process, two regularization terms[6] are added to the objective function in Equation 4 as follows:

[6]We use the Frobenius norm as it is commonly used to formulate the matrix factorization problem. It allows to highly penalize high values in the case of the regularization parameters.

$$\underset{M_U^d, M_T^d}{\arg\min}\,\mathcal{L} \quad = \quad \underset{M_U^d, M_T^d}{\arg\min}\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(M_{u_i,t_j}^d - M_{u_i}^{'d} \times M_{t_j}^d)^2 + \frac{\lambda}{2}(\|M_U^d\|_F^2 + \|M_T^d\|_F^2) \quad (5)$$

where $\lambda > 0$ is a regularization weight.

The optimization problem in Equation 5 minimizes the sum-of-squared-errors between observed and predicted weightings. The gradient descent algorithm can be applied to find a local minimum in feature vectors $M_{u_i}^d$ and $M_{t_j}^d$, where we have:

$$\frac{\partial \mathcal{L}}{\partial M_{u_i}^d} = \sum_{j=1}^{n} I_{ij}(M_{u_i}^{'d} \times M_{t_j}^d - M_{u_i,t_j}^d)M_{t_j}^d + \lambda M_{u_i}^d \quad (6)$$

$$\frac{\partial \mathcal{L}}{\partial M_{t_j}^d} = \sum_{i=1}^{m} I_{ij}(M_{u_i}^{'d} \times M_{t_j}^d - M_{u_i,t_j}^d)M_{u_i}^d + \lambda M_{t_j}^d \quad (7)$$

Once we have computed the factorized user latent features and tag latent features matrices, we can predict missing values using $M_U^{'d} \times M_T^d$. Then, we consider that:

Proposition 1. *The row that corresponds to the query issuer in the predicted matrix $M_U^{'d} \times M_T^d$ corresponds to his/her PerSaDoR for the considered document. A PerSaDoR is represented as a weighted vector of terms.*

This process is shown in Step 6 of Figure 3. In the next section, we describe our method to compute a ranking score for documents, w.r.t. their PerSaDoR, their textual content, and the query.

4.4. Ranking documents using PerSaDoR

In the previous sections, we have formalized a PerSaDoR of a document that matches the query of a user. The PerSaDoRs have to be matched to the query for quantifying their similarities while also considering the textual content of the documents. Therefore, we propose to compute ranking scores for documents using one of the following ranking functions:

1. A *Query Based Ranking Function* (QBRF), where the personalized ranking score of a document d that match a query q issued by a user u is computed as follows:

$$Rank(d, q, u) = \gamma \times Sim(\overrightarrow{q}, \overrightarrow{S_{d,u}}) + (1 - \gamma) \times SES(\overrightarrow{d}) \quad (8)$$

2. A *Profile Based Ranking Function* (PBRF), following the same idea as in [13, 34, 40, 42]. The personalized ranking score of a document d that matches a query q issued by a user u is computed as follows:

$$Rank(d, q, u) = \gamma \times Sim(\overrightarrow{p_u}, \overrightarrow{S_{d,u}}) + (1 - \gamma) \times SES(\overrightarrow{d}) \quad (9)$$

where, in both formulas, γ is a weight that satisfies $0 \leq \gamma \leq 1$, $SES(\overrightarrow{d})$ is the Search Engine Score (SES) given to the document d, e.g., we use the *Apache Lucene* search engine in our implementation[7] [32], $\overrightarrow{S_{d,u}}$ is

[7] https://lucene.apache.org/core/5_3_1/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

the PerSaDoR of the document d according to the user u, and $\vec{p_u}$ is the user profile constructed following Definition 3.

Inspired by the Vector Space Model, queries, documents, and PerSaDoRs are modeled as vectors. Therefore, we compute the similarities between these vectors using the cosine measure as follows:

$$Sim(\overrightarrow{q}, \overrightarrow{S_{d,u}}) = \frac{\overrightarrow{q} \bullet \overrightarrow{S_{d,u}}}{|\overrightarrow{q}| \times |\overrightarrow{S_{d,u}}|}, \qquad Sim(\overrightarrow{p_u}, \overrightarrow{S_{d,u}}) = \frac{\overrightarrow{p_u} \bullet \overrightarrow{S_{d,u}}}{|\overrightarrow{p_u}| \times |\overrightarrow{S_{d,u}}|} \tag{10}$$

Finally, note that our method is applied on the top 10.000 documents obtained after an initial run of a query on the constructed textual index. Thus, this list is re-ranked 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. Then, the top ranked documents are formatted for presentation to the user.

In the next section, we provide a complexity analysis of our approach and we show the execution time needed to factorize a number of documents, motivating the choice of running the processes on the fly, as mentioned before.

4.5. Complexity analysis

The main computation effort for generating a PerSaDoR of a document is spent in building the Users-Tags matrix and factorize it (Steps 1 to 5 in Figure 3). The time complexity needed for building a Users-Tags matrix is $O(\mid U_d \mid \times log(\mid U_d \mid))$, which corresponds to rank users for selecting the most representative ones (step 2 in Figure 3). For factorizing the matrix, the main computation of the gradient descent algorithm is evaluating the objective function $\mathcal{L}$ in Equation 5 and its derivatives in Equations 6 and 7. As pointed in [29], 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 (see Figure 4). Therefore, the computational complexity of evaluating the objective function $\mathcal{L}$ is $O(\rho)$, where ρ is the number of nonzero entries in the Users-Tags matrix. Also, the computational complexity for the derivatives $\frac{\partial \mathcal{L}}{\partial M^d_{u_i}}$ and $\frac{\partial \mathcal{L}}{\partial M^d_{t_j}}$ of Equations 6 and 7 respectively are the same which is $O(\rho)$. Thus, the total computational complexity in one iteration of the gradient descent algorithm is $O(\rho)$. 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 (on average $i \backsimeq 15$ in our evaluations). Finally, for computing a PerSaDoR of a given document, the time complexity is estimated to:

$$O(\mid U_d \mid \times log(\mid U_d \mid) + i \times \rho) \tag{11}$$

As a last step, the computational complexity for evaluating a query q that matches m documents is estimated to be:

$$O(m \times [\mid U_d \mid \times log(\mid U_d \mid) + i \times \rho]) \tag{12}$$

Since i, ρ and $\mid U_d \mid$ are expected to be low values due to the sparse nature of folksonomies, we can say that the complexity scales linearly with the number of retrieved documents which indicates that this approach

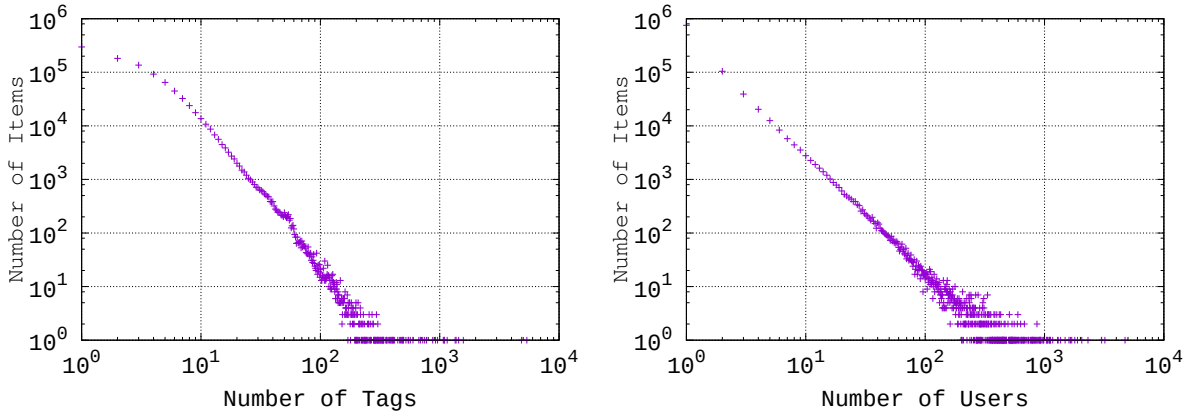


Figure 4: Distribution over documents.

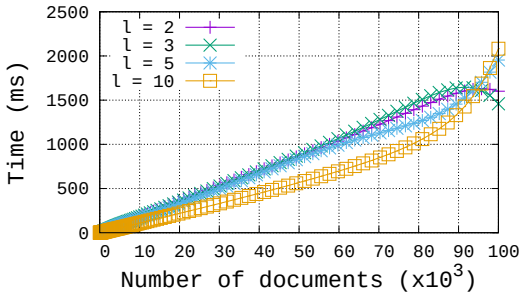
can be applied to very large datasets. By using parallel computation, we can easily and considerably reduce the execution time even more. This is part of our future work.

As an illustration, Figure 5 shows the execution time needed for processing queries according to the number of matched documents w.r.t. several parameters. These latter are: (i) l, the number of latent dimensions with which we perform the factorization, and (ii) k, the number of related users chosen to build the Users-Tags matrix. The queries and the users were randomly selected 10 times independently, and we report the average results each time. As depicted in Figure 5, none of these parameters have an impact on the execution time. This latter still scales linearly with the number of documents. Note that the average execution time of the factorization of a single Users-Tags matrix in our experiments was about $15\mu s$. The factorization process was on average converging after 15 iterations. The results are obtained on a MacBook Pro with a 2.8GHz Intel Core i7 CPU and 4GB 1333MHz DDR3 of RAM, running MacOS X Lion v10.7.4.

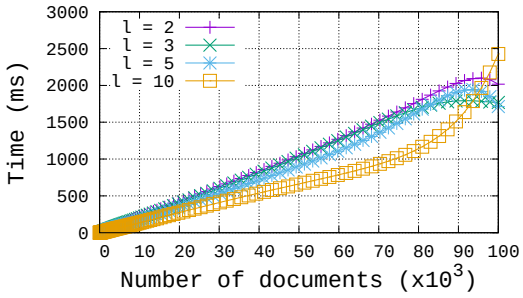
5. Experimental Evaluation

To demonstrate the interest of our approach, we have performed extensive evaluations over a large dataset and checked different aspects of the approach, as we will see in the next sections. In this section, we describe the dataset used, the evaluation methodology, and the metrics used to evaluate our approach. Note that our approach has been implemented using the *Apache Lucene* search engine.[8]

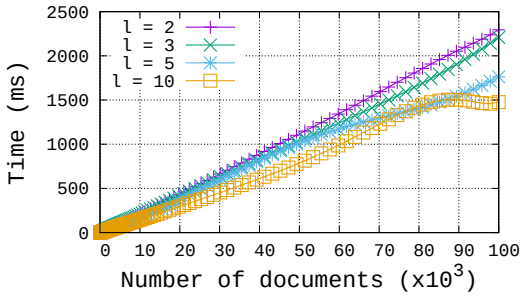
[8] `http://lucene.apache.org/`



(a) Top 1 related user.



(b) Top 3 related user.



(c) Top 5 related user.

Figure 5: Execution time for processing queries with respect to the number of documents that they match.

5.1. Dataset

We have selected a *delicious*[9] dataset to perform our evaluations. This dataset is public, described and analyzed in [41][10]. The interest of using such data instead of crawled data is to work on widely accepted data by the community. This also allows reducing the risk of noise, to reproduce the evaluations by others, and to compare our approach to other approaches on "standardized datasets".

Before the experiments, we performed mainly five data pre-processing tasks: (1) Several annotations are too personal or meaningless, such as "toread", "Imported IE Fa-vorites", "system:imported", etc. We remove some of them manually. (2) Although the annotations from *delicious* are easy to read and understand by users, they are not designed for machine use. For example, some users may concatenate several words to form an annotation such as "java.programming" or "java/programming". We tokenize this kind of annotations before using them in the experiments. (3) 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. In the same time, the system records the relations between stemmed terms and original terms. (4) We downloaded all the available Web pages while removing those, which are no longer available using the *cURL* command line

[9] `http://www.delicious.com/`

delicious is a social bookmarking Web service for storing, sharing, and discovering Web bookmarks.

[10] `http://data.dai-labor.de/corpus/delicious/`

Table 3: 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

tool.[11] (5) Finally, we removed all the non-english Web pages. This operation was performed using *Apache Tika* toolkit.[12]

Table 3 gives a description of the resulted dataset after cleansing. This dataset has the same properties as the initial dataset. In other words, it is very sparse and follows a long tail distribution [21, 41], i.e., most URLs are tagged by only a handful of users, and few users only use many tags.

5.2. Offline evaluation methodology

Setting up evaluations for personalized search is a challenge since relevance judgements can only be assessed by end-users themselves [13]. This is difficult to achieve at a large scale. However, different efforts [4, 24] state that the tagging behavior of a user of folksonomies closely reflects his/her behavior of search on the Web. In other words, if a user tags a resource r with a tag t, he/she will choose to access the resource r if it appears in the result obtained by submitting t as a query to the search engine. Thus, we can easily state that any bookmark (u, t, r) that represents a user u who bookmarked a resource r with tag t, can be used as a test query for evaluations. The main idea of these experiments is based on the following assumption:

Assumption 1. *For a personalized query $q = \{t\}$ issued by user u with query term t, the relevant documents are those tagged by u with t.*

Hence, in the off-line study, for each evaluation, we randomly select $2,000$ pairs (u, t), which are considered to form a personalized query set. For each corresponding pair (u, t), we remove all the bookmarks $(u, t, r) \in \mathbb{F}, \forall r \in R$ in order to not promote the resources r 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 as explained throughout this paper, where documents are indexed using *Apache Lucene.* Then, according to the previous assumption, we consider that the relevant documents are those tagged by u using tags of q to assess the obtained results.

5.3. Evaluation metrics

We use the *Mean Average Precision (MAP)* and the *Mean Reciprocal Rank (MRR)*, two performance measures that take into account the ranking of relevant documents. (a) Starting from the obtained list of search results, the average-precision is computed. Then, the MAP is computed over the $2,000$ queries. (b)

[11] All the Web pages that return an http error code were considered to be unavailable.

[12] `http://tika.apache.org/1.1/api/org/apache/tika/language/LanguageIdentifier.html#getLanguage()`

Table 4: Default values of the parameters for their evaluation.

Parameter	**Value**	**Remark**
γ	1	To better estimate the impact of the PerSaDoR on the other parameters
α	0	To better discriminate between users while varying the other parameters
Similarity	Cosine	/
Top users	2	/
Dimension	5 or 10	/
λ	0.02	/

The MRR is computed as the multiplicative inverse of the rank of the first correct answer, averaged over the $2,000$ queries. MAP and MRR are defined as:

$$MAP = \frac{1}{|q|}\sum_{j=1}^{|q|}\frac{\sum_{r=1}^{N}(P(r)\times rel(r))}{|R_q|}, \qquad MRR = \frac{1}{|q|}\sum_{i=1}^{q}\frac{1}{rank_i} \tag{13}$$

where $P(r)$ is the precision at cut-off k in the list, $rel(r)$ is an indicator that equals to 1 if the resource at rank k is relevant, 0 otherwise. $|q|$ is the total number of queries and $rank_i$ is the rank of the first relevant document in the retrieved list of documents that match the query q returned by the system. Finally, $|R_q|$ is the number of relevant resources for q .

In the evaluation, the random selection of the $2,000$ queries was carried out 10 times independently, and we report the average results. In all the evaluations, we refer to our approach as *"PerSaDoR QBRF"* for the first ranking function (i.e., Equation 8), and *"PerSaDoR PBRF"* for the second ranking function (i.e., Equation 9).

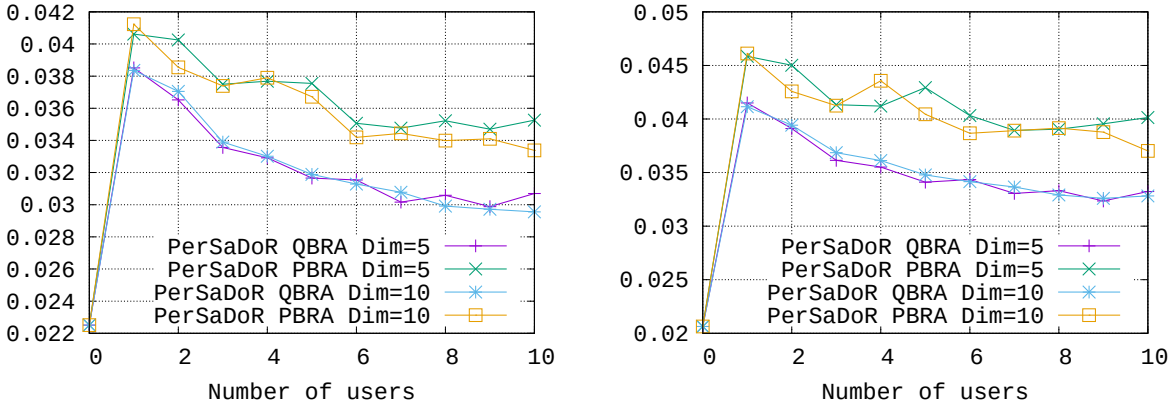
5.4. Estimation of the parameters

Our approach possesses several parameters that can be tuned. While studying the impact of a parameter, we fix each time the others to the values described in Table 4. Note that each time, we give the results obtained using: (i) two different dimensions for the factorization process (5 and 10), and (ii) our two ranking functions.

5.4.1. Impact of the number of users (k)

The results obtained while varying the number of users are illustrated in Figure 6. The results show that optimal results are obtained while selecting 1 or 2 related users depending on the ranking function and the retrieval process used. Adding more users decreases significantly the performance. This is due to the fact that the filtered out users have inappropriately annotated documents and are socially far from the query

issuer. These users represent the irrelevant users that we would like to set aside. Thus, these results show the effectiveness of the ranking function proposed in Section 4.2.1.



(a) Mean Average Precision.

(b) Mean Reciprocal Rank.

Figure 6: Impact of the number of users.

5.4.2. Impact of the social proximity part (α)

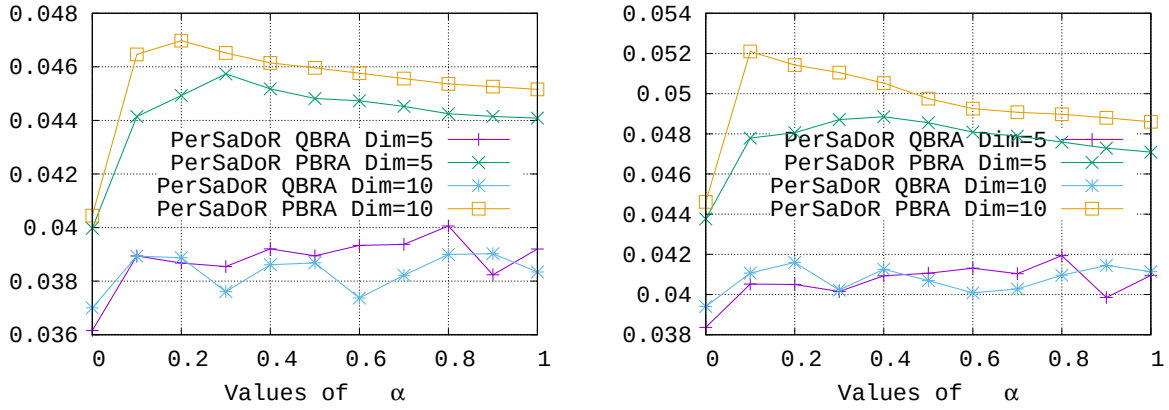
The results obtained while varying this parameter are illustrated in Figure 7. This parameter allows to control the social proximity and the document proximity parts while computing the ranking scores for users in Equation 1. The obtained results show that the optimal performance is obtained for $\alpha \in [0.1, 0.4]$, improving the MAP and MRR by 3% and 4% for respectively the QBRF and PBRF ranking functions. On the one hand, considering only the social proximity part does not provide good performance ($\alpha = 0$). This is due to the fact that there are many users who have annotated relevant documents with relevant tags, and who don't share affinity with the query issuer. On the other hand, considering only the document proximity part does not necessarily provide a good retrieval performance ($\alpha = 1$). This is due to the fact that we are not taking into account the social dimension for discriminating between users.

5.4.3. Impact of the similarity measure

The results obtained using different similarity measures are illustrated in Figure 8. Clearly, the cosine similarity measure provides the best retrieval performance by allowing being more efficient in discriminating between users. This is certainly due to the fact that the cosine measure takes into account the importance of each tag for each user while computing similarities. The other similarities are purely statistical since they consider only the number of tags (in common) without estimating the importance of each of these tags.

5.4.4. Impact of the PerSaDoR score (γ)

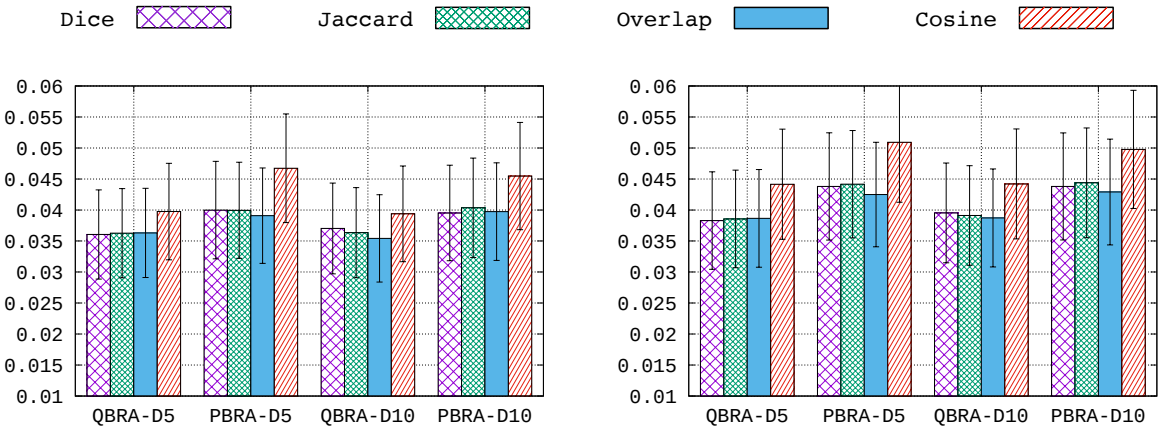
The results obtained by tuning this parameter are illustrated in Figure 9. The optimal value is obtained for $\gamma \in [0.6, 0.9]$, a value which we consider as a trade-off between the personalized and the non-personalized



(a) Mean Average Precision.

(b) Mean Reciprocal Rank.

Figure 7: Impact of α.



(a) Mean Average Precision.

(b) Mean Reciprocal Rank.

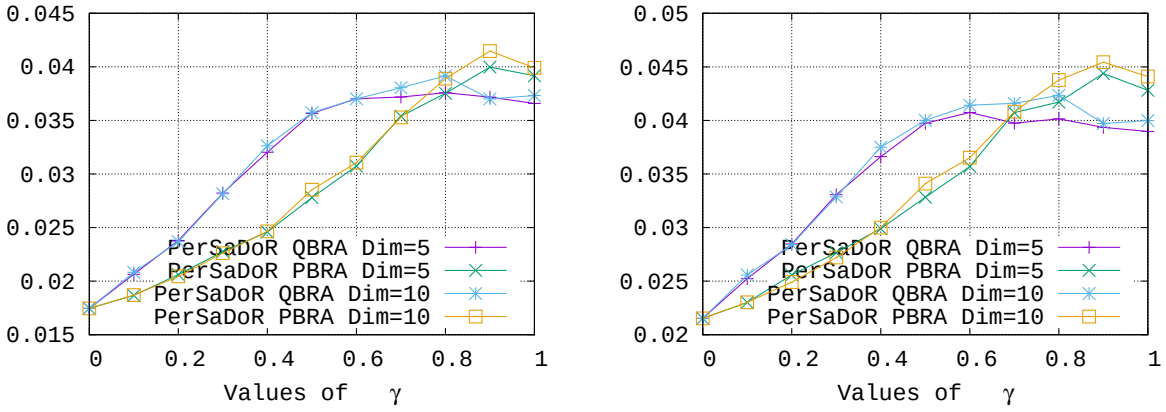
Figure 8: Impact of the similarity measure. 95% confidence intervals are shown.

parts. This also shows that our method is effectively improving the performance by improving MAP from 0.0155 to 0.041 and MRR from 0.0205 to 0.0451 for $\gamma = 0.9$. This represents an improvement of almost 100%.

In the next sections, we describe two types of evaluations that we have performed on our approach: (i) comparison with baselines, and (ii) a user survey. These evaluations are expected to provide a full picture of the benefits and limitations of the proposed approach.

6. Comparison with baselines

Our objective here is to analyze how well our approach meets the users' information needs compared with other state of the art approaches. Our approach is evaluated using the optimal values computed in the previous section while using five dimensions in the factorization process and our two ranking functions as



(a) Mean Average Precision.

(b) Mean Reciprocal Rank.

Figure 9: Impact of γ.

explained in Section 4.4. Note that the comparison is performed on a different test set of queries than those used in the training set to learn the optimal parameters. 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. These baselines are summarized and described in Table 5.

Table 5: Summary of the baselines.

		Baseline	Description
Non-personalized approaches	**1**	**SPR [2]**	SocialPageRank (SPR) captures the popularity (quality) of web pages from the web users' perspective in a folksonomy. We use the SPR score for ranking of web pages by treating it as independent evidence using the following formula: $Rank(u, q, d) = \gamma \times SPR(d) + (1 - \gamma) \times SES(\overrightarrow{d})$
	2	**Dmitriev06 [16]**	Briefly, the authors propose to combine the annotations with the content and anchor text of documents to produce a new index. Currently, for retrieval and ranking purposes annotations are treated as if they were textual content of documents. We implemented this approach using the Apache Lucene search engine.
	3	***B*L-Q**	This approach use a query based ranking function as described in Equation 8. However, we use a social representation of documents based on all their annotations weighted using the *tf-idf* measure.
	4	**Lucene**	This approach is the Lucene naive function where all the parameters have been set to their default values [32].
	5	**LDA-Q**	This approach use LDA [5] for modeling queries and documents. Then, for each document that matches a query, we compute a similarity between its topic and the topic of the query using the cosine measure (inferred using the previous constructed model). The obtained value is merged with the textual ranking score as in Equation 8.
Personalized approaches	**6**	**Xu08 [42]**	This approach use a profile based ranking function where documents and users are weighted using the *tf-idf*.
	7	**Noll07 [34]**	The approach considers only a user interest matching between a user and a document. It does not make use of the user and document length normalization factors, and only uses the user tag frequency values. The authors normalize all document tag frequencies to 1 since they want to give more importance to the user profile.
	8	**tf-if [40]**	This approach is an adaptation of **[34].** The main difference is that tf-if incorporates both the user and document tag distribution global importance factors, following the VSM principle.
	9	**Semantic Search [3]**	This approach ranks documents by considering users that hold similar content to the query, i.e., users who used at least one of the query terms in describing their content.
	10	**LDA-P**	We also propose an approach based on LDA to model users and documents. Then, for each document that matches a query, we compute a similarity between its topic and the topic of the user profile using the cosine measure (inferred using the previous constructed model). The obtained value is merged with the textual ranking score as in Equation 9.
Note that we use a Java implementation of the Latent Dirichlet Allocation (LDA) using Gibbs Sampling for Parameter Estimation and Inference[13]. In each execution, we use the default values proposed by this implementation, i.e., $\alpha = 0.5$, $\beta = 0.1$, $topics = 100$, and a number of most likely words for each topic equal to 20.			

6.1. Results Analysis

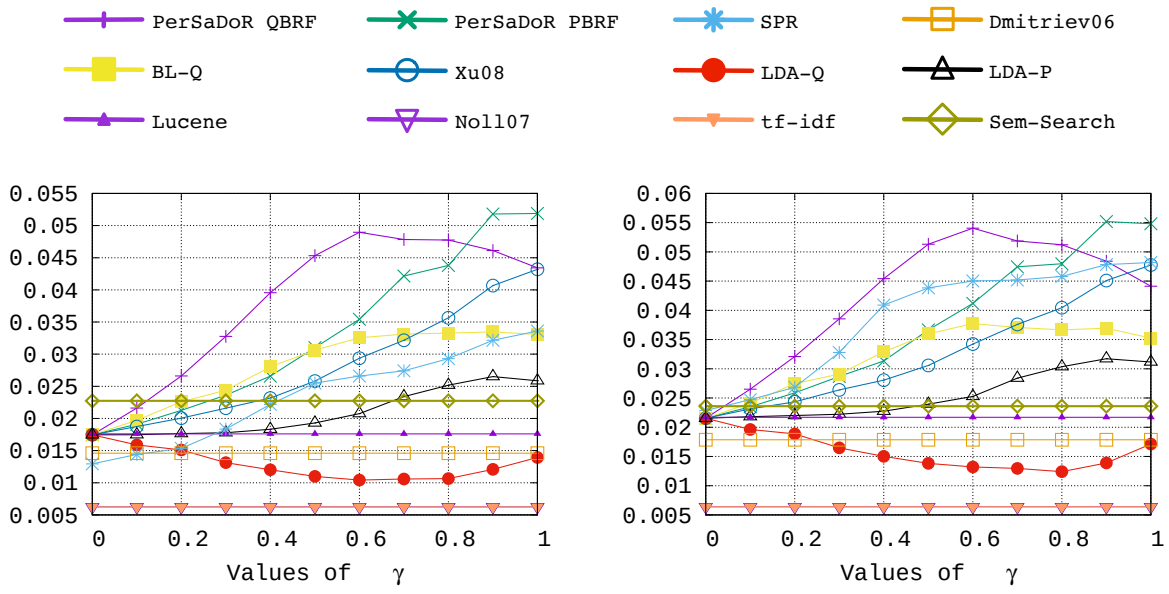
The results of the comparison are illustrated in Figure 10, while varying γ.

6.1.1. PerSaDoR vs non-personalized approaches

First, we wanted to ensure that our approach is providing an added value compared to the non-personalized methods. As illustrated in Figure 10, the obtained results show clearly that our approach is much more efficient than all the non-personalized approaches for all values of γ. Therefore, 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 decrease their performance for high values of γ. This is due to the fact that they are not designed for personalized search, since these approaches fail to discriminate between users.

6.1.2. PerSaDoR vs personalized approaches

Here, the obtained results also show that our approach is much more efficient than all the personalized approaches for all values of γ (except for $\gamma = 0$, where Semantic Search gives better results). Especially, our approach outperform the LDA-P approach and the Xu08 approach, which we consider as the closest works to ours. 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 don't share tags with users since in our experiment we remove the triplets that associate the user, the query terms and documents.



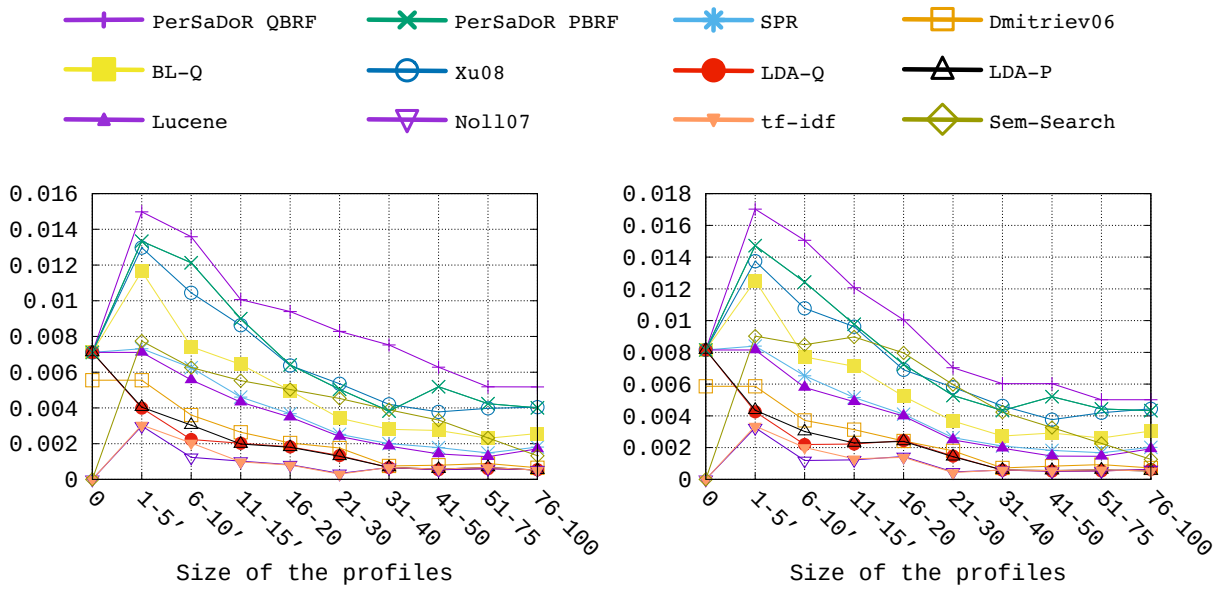
(a) Mean Average Precision.

(b) Mean Reciprocal Rank.

Figure 10: Comparison with the baselines while varying γ and using the optimal values of the parameters.

6.2. Performance on different queries

In this section, we study the ability of our approach to achieve a good performance even if the users have annotated documents with few terms. Therefore, to do so, we propose to compare our approach with the other baselines while following the same evaluation process as described in Section 5.2. We select $2,000$ query pairs (u, t) based on the number of tags the users used in their tagging actions. The query pairs are grouped into 10 classes: "0", "1-5", "6-10", "11-15", "16-20", "21-30", "13-40", "41-50", "51-75", and "76-100", denoting how many tags users have used in their tagging actions, e.g., class "1-5" is composed with users who have a profile length between 1 and 5. Note that we select the optimal values of the parameters of the PDSV framework as discussed in Section 5.4, while fixing $\gamma = 0.5$ for all the approaches.



(a) Mean Average Precision.

(b) Mean Average Precision.

Figure 11: Performance comparison on different queries, while fixing $\gamma = 0.5$.

The experimental results are shown in Figure 11 over the 10 classes of queries. The obtained results show that the PerSaDoR approach outperforms almost all the baseline approaches for all the queries. We also report that even if a user doesn't annotate a Web page, the PerSaDoR approach still can improve the search quality comparing to other approaches. This is due to the fact that reasonable weightings are predicted in the Users-Tags matrix since the explicit feedback of the closest users is used to compute a PerSaDoR of each document that potentially match a query. These results show the effectiveness of the PerSaDoR approach in the context of sparse data.

The results of this offline evaluation show that our approach is much more efficient than all the baselines even if the query issuer doesn't annotate a Web page. Especially, our approach outperforms all the personalized approaches which we consider as the closest to our contribution. Hence, we conclude that the personalization efforts introduced by our approach brings a considerable improvement for the search quality.

Figure 12: User survey Web page.

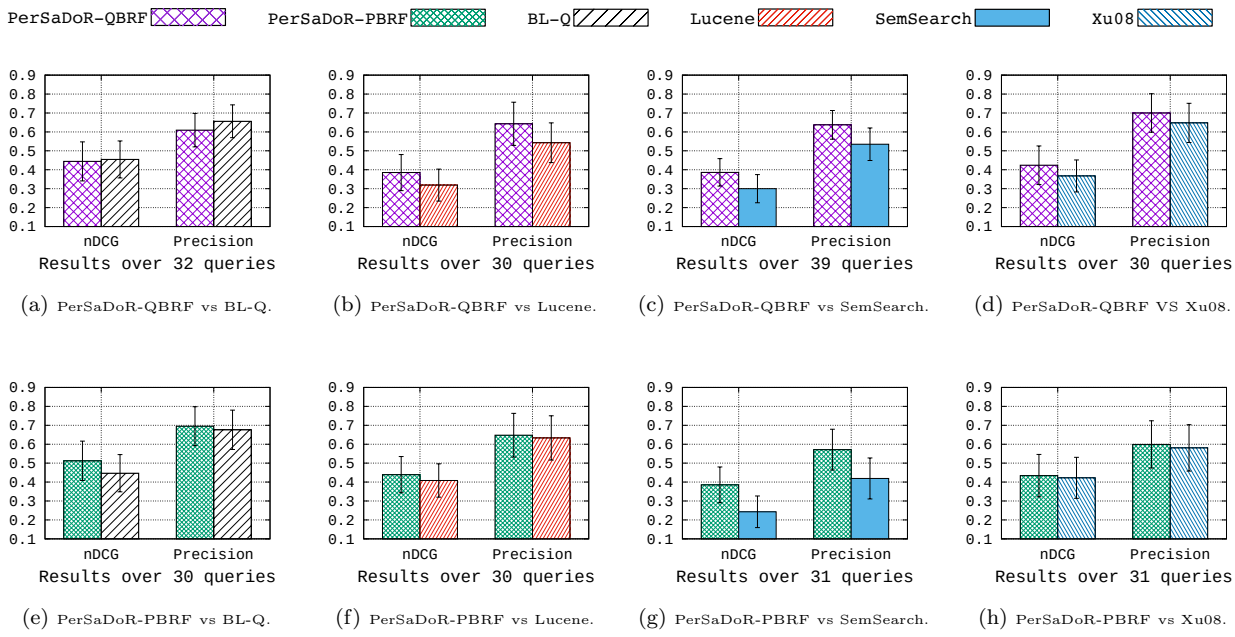
Finally, we note that in this offline evaluation, the best performance is obtained while using QBRF and choosing one or two of the most related users to the query issuer. However, these results should be reinforced using an online evaluation to give a better overview of the performance through a user survey. This is detailed in the next section.

7. User survey

For the user study, we have used our *delicious* dataset from which we have selected 335 pairs of queries and users. These users are considered as query initiators and have used all the selected query tags at least once on the same document. We then run the queries using our approach and the baselines that performed the best in the offline evaluation. At each iteration, the user is presented with two lists of 10 ranked documents generated using: (i) our approach and (ii) a randomly selected baseline algorithm. Note that, at this stage, end-users don't know which approach is ours and which one is the baseline. For all approaches, γ was set to 0.5.

In the assessment phase, 39 volunteers participated to judge the relevance of the results. Each volunteer (who is considered as a query initiator) was shown, in addition to the results for the query from the pool: (i) the documents from the query initiator that contain at least one of the query tags and (ii) the tags he/she used in his/her tagging actions. This is to help the volunteers to understand the personal context of the (real) query initiators as well as their interests. This way, we intend to overcome the aforementioned problem of subjectively assessing the result quality with the eyes of the query initiator.

Once a list is presented to a participant, he/she marks each result as: very relevant, relevant, or irrelevant w.r.t the context of the (real) query initiator. This process is performed by the evaluator without knowing which algorithm has generated the lists. Figure 12 shows the interface obtained by the users when they participated to the survey. This interface contains (i) the tags used by the user in his/her tagging actions



(a) PerSaDoR-QBRF vs BL-Q. (b) PerSaDoR-QBRF vs Lucene. (c) PerSaDoR-QBRF vs SemSearch. (d) PerSaDoR-QBRF VS Xu08.

(e) PerSaDoR-PBRF vs BL-Q. (f) PerSaDoR-PBRF vs Lucene. (g) PerSaDoR-PBRF vs SemSearch. (h) PerSaDoR-PBRF vs Xu08.

Figure 13: User survey: The precision of the search results for different algorithms measured by nDCG@10 and P@10.

(in the top right part), (ii) the documents he/she tags with the query terms (in the right part), and (iii) the two lists of results to be judged after the query was issued.

The quality of each result was measured by the normalized discount cumulative gain (nDCG@10) and by precision at 10 (P@10), averaged over the set of judged queries. For DCG calculation, we used gains (2,1,0) for the three relevance levels respectively, and the discount function used was $DCG = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)}$. Normalization (nDCG) was done by dividing the DCG value with an ideal DCG value calculated as all results are highly relevant. For the P@10 calculation, we considered any positive judgment as relevant. The obtained results are shown in Figure 13 as measured by NDCG@10 and P@10.

The main outcome of the survey can be summarized as follows: (i) this user survey confirms, to some extent, the results obtained in the offline evaluation since the PerSaDoR approaches outperform the selected baselines. (ii) The BL-Q approach, even if it is a non-personalized approach, has been judged to be more efficient than the PerSaDoR-QBRF approach. (iii) The advantage observed by the PerSaDoR approaches is not as important. Actually, several participants mentioned the difficulty in judging the relevance of the queries, mostly because of unfamiliarity with the users they are related to. (iv) We believe that the best performance is provided by the PerSaDoR-PBRF approach since it outperforms the baselines. This remark should be confirmed by evaluating the two PerSaDoR approaches together on the same queries as this has been done with the baselines.

As a conclusion for this evaluation, we notice that there are several substantial differences between the two evaluation methods. Both methods confirm the significant contribution of the personalization introduced

in the representation of documents using the PerSaDoR approach, and the superiority of using it for ranking purposes. However, the results obtained in the offline evaluation show the superiority of the PerSaDoR-QBRF approach over the PerSaDoR-PBRF which is not what we observed in the user survey. Also, although the superiority of the PerSaDoR approach has been clearly observed in the offline evaluation, the user survey showed some subtlety regarding this superiority, i.e., the superiority of the PerSaDoR approach is not so obvious in the user survey. Subjective constraints need to be taken into account in this process like the one mentioned before. These results have to be confirmed eventually by a more realistic evaluation where we consider users with their own accounts to be fully aware of the context.

8. Conclusion and future work

This paper discusses a contribution to the area of IR modeling while leveraging the social dimension of the Web. We propose a Personalized Social Document Representation framework (PerSaDoR), an attempt to use social information to enhance, improve and provide a personalized representation of documents to users. When a user submits a query, we construct, on the fly, a PerSaDoR of all documents that potentially match the query based on other user's experience (while considering both users that are socially close to the query issuer and relevant to documents). Then, we rank these documents with respect to one of the two ranking functions that we proposed. The complexity analysis that we have performed shows that personalizing the IR process at this stage is possible with relatively an acceptable execution time. Also, the extensive experiments that we have performed on a *delicious* dataset show the benefit of such an approach compared to the state of the art.

Even with the interest of the proposed method, there are still possible improvements that we can bring. We are investigating the possibility of deploying our method in a distributed setting where data are often distributed on different clusters of a data center. We are also investigating ways to add social regularization terms to the objective function of the matrix factorization in order to model other behaviours of users. The temporal dimension of social users' behavior has not been investigated yet; this is also part of our future work to improve our proposal. Finally, we are currently working on plugging our previous work of social query expansion [9] into a common prototype. PerSaDoR has been developed and integrated to the LAICOS [6] platform.

Acknowledgement

We would like to thank the anonymous reviewers for their insightful comments on the paper, as these comments led us to an improvement of the work. We also would like to thank Ian Davison, English Instructor at Zayed University and IELTS Examiner associated with the British Embassy in the UAE for his proof reading of the paper. This work has been primarily completed while Dr Bouadjenek and Dr Hacid were researchers at Bell Labs France, Centre de Villarceaux, 91620 Nozay.

[1] S. Amer-Yahia, M. Benedikt, L. V. S. Lakshmanan, and J. Stoyanovich. Efficient network aware search in collaborative tagging sites. *Proc. VLDB Endow.*, 1(1):710–721, August 2008.

[2] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 501–510, New York, NY, USA, 2007. ACM.

[3] 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*, pages 501–506. IEEE Computer Society, 2008.

[4] K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. Can all tags be used for search? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 193–202, New York, NY, USA, 2008. ACM.

[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 *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '13, pages 1446–1449, New York, NY, USA, 2013. ACM.

[7] M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub. SoPRa: A New Social Personalized Ranking Function for Improving Web Search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 861–864, New York, NY, USA, 2013. ACM.

[8] M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub. Social networks and information retrieval, how are they converging? a survey, a taxonomy and an analysis of social information retrieval approaches and platforms. *Information Systems*, 56:1–18, 2016.

[9] M. R. Bouadjenek, H. Hacid, M. Bouzeghoub, and J. Daigremont. Personalized Social Query Expansion Using Social Bookmarking Systems. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1113–1114, New York, NY, USA, 2011. ACM.

[10] M. R. Bouadjenek, H. Hacid, M. Bouzeghoub, and A. Vakali. Using Social Annotations to Enhance Document Representation for Personalized Search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 1049–1052, New York, NY, USA, 2013. ACM.

[11] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, September 2002.

[12] D. Carmel, H. Roitman, and E. Yom-Tov. Social bookmark weighting for search and recommendation. *The VLDB Journal*, 19:761–775, December 2010.

[13] D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'el, I. Ronen, E. Uziel, S. Yogev, and S. Chernov. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1227–1236, New York, NY, USA, 2009. ACM.

[14] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *International Journal of Computer Vision*, 80(1):125–142, 2008.

[15] S.-Y. Chen and Y. Zhang. Improve web search ranking with social tagging. *MSM*, 2009.

[16] P. A. Dmitriev, N. Eiron, M. Fontoura, and E. Shekita. Using annotations in enterprise search. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 811–817, New York, NY, USA, 2006. ACM.

[17] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th international conference on World Wide Web*, WWW '10, pages 331–340, New York, NY, USA, 2010. ACM.

[18] Q. Du, H. Xie, Y. Cai, H. fung Leung, Q. Li, H. Min, and F. Wang. Folksonomy-based personalized search by hybrid user profiles in multiple levels. *Neurocomputing*, 204:142 – 152, 2016. Big Learning in Social Media AnalyticsContaining a selection of papers from the 2014 International Conference on Security, Pattern Analysis, and Cybernetics (ICSPAC2014).

[19] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11(4), April 2005.

[20] X. He, M. Gao, M.-Y. Kan, Y. Liu, and K. Sugiyama. Predicting the popularity of web 2.0 items based on user comments. In *Proceedings of the 37th International ACM SIGIR Conference on Research; Development in Information Retrieval*, SIGIR '14, pages 233–242, New York, NY, USA, 2014. ACM.

[21] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search? In *Proceedings of the international conference on Web Search and Web Data Mining*, WSDM '08, pages 195–206, New York, NY, USA, 2008. ACM.

[22] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, 2006.

[23] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[24] B. Krause, A. Hotho, and G. Stumme. A comparison of social bookmarking with traditional search. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 101–113, Berlin, Heidelberg, 2008. Springer-Verlag.

[25] H. Kumar, S. Lee, and H.-G. Kim. Exploiting social bookmarking services to build clustered user interest profile for personalized search. *Information Sciences*, 281:399 – 417, 2014. Multimedia Modeling.

[26] H. Lai, Y. Pan, C. Liu, L. Lin, and J. Wu. Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Transactions on Computers*, 62(6):1221–1233, June 2013.

[27] C. Li, L. Lin, W. Zuo, S. Yan, and J. Tang. Sold: Sub-optimal low-rank decomposition for efficient video segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 5519–5527, June 2015.

[28] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 203–210, New York, NY, USA, 2009. ACM.

[29] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 931–940, New York, NY, USA, 2008. ACM.

[30] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web Search and Data Mining*, WSDM '11, pages 287–296, New York, NY, USA, 2011. ACM.

[31] B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme. Evaluating similarity measures for emergent semantics of social tagging. In *Proceedings of the 18th international conference on World Wide Web*, WWW '09, pages 641–650, New York, NY, USA, 2009. ACM.

[32] M. McCandless, E. Hatcher, and O. Gospodnetic. *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA, 2010.

[33] M.-T. Nguyen and M.-L. Nguyen. *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, chapter SoRTESum: A Social Context Framework for Single-Document Summarization, pages 3–14. Springer International Publishing, Cham, 2016.

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

[35] Z. Saoud and S. Kechid. Integrating social profile to improve the source selection and the result merging process in distributed information retrieval. *Information Sciences*, 336:115 – 128, 2016.

[36] M. Servajean, R. Akbarinia, E. Pacitti, and S. Amer-Yahia. Profile diversity for query processing using user recommendations. *Information Systems*, 48:44 – 63, 2015.

[37] O. Shafiq, R. Alhajj, and J. G. Rokne. On personalizing web search using social network analysis. *Information Sciences*, 314:55 – 76, 2015.

[38] S. Siersdorfer, P. Kemkes, H. Ackermann, and S. Zerr. Who with whom and how?: Extracting large social networks using search engines. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1491–1500, New York, NY, USA, 2015. ACM.

[39] T. Takahashi and H. Kitagawa. A ranking method for web search using social bookmarks. In *Proceedings of the 14th International Conference on Database Systems for Advanced Applications*, DASFAA '09, pages 585–589, Berlin, Heidelberg, 2009. Springer-Verlag.

[40] D. Vallet, I. Cantador, and J. M. Jose. Personalizing web search with folksonomy-based user and document profiles. In *Proceedings of the 32nd European conference on Advances in Information Retrieval*, ECIR'2010, pages 420–431, Berlin, Heidelberg, 2010. Springer-Verlag.

[41] R. Wetzker, C. Zimmermann, and C. Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Proceedings of the ECAI 2008 Mining Social Data Workshop*, pages 26–30. IOS Press, 2008.

[42] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 155–162, New York, NY, USA, 2008. ACM.

[43] Z. Xu, T. Lukasiewicz, and O. Tifrea-Marciuska. Improving personalized search on the social web based on similarities between users. In U. Straccia and A. Calì, editors, *Scalable Uncertainty Management*, volume 8720 of *Lecture Notes in Computer Science*, pages 306–319. Springer International Publishing, 2014.

[44] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Towards improving web search by utilizing social bookmarks. In *Proceedings of the 7th international conference on Web engineering*, ICWE'07, pages 343–357, Berlin, Heidelberg, 2007. Springer-Verlag.

[45] D. Yang, D. Zhang, Z. Yu, Z. Yu, and D. Zeghlache. Sesame: Mining user digital footprints for fine-grained preference-aware social media search. *ACM Trans. Internet Technol.*, 14(4):28:1–28:24, December 2014.

[46] J. Yu, Y. Rui, and B. Chen. Exploiting click constraints and multi-view features for image re-ranking. *IEEE Transactions on Multimedia*, 16(1):159–168, Jan 2014.

[47] J. Yu, Y. Rui, and D. Tao. Click prediction for web image reranking using multimodal sparse coding. *IEEE Transactions on Image Processing*, 23(5):2019–2032, May 2014.

[48] J. Yu, D. Tao, M. Wang, and Y. Rui. Learning to rank using user clicks and visual features for image retrieval. *IEEE Transactions on Cybernetics*, 45(4):767–779, April 2015.

[49] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Transactions on Image Processing*, 24(12):4766–4779, Dec 2015.

[50] X. Zhang, L. Yang, X. Wu, H. Guo, Z. Guo, S. Bao, Y. Yu, and Z. Su. sdoc: exploring social wisdom for document enhancement in web mining. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 395–404, New York, NY, USA, 2009. ACM.