

A Multi-factor Tag-Based Personalized Search

Frederico Durao¹, Ricardo Lage¹, Peter Dolog¹, and Nilay Coskun²

¹ Intelligent Web and Information Systems, Aalborg University, Computer Science Department
Selma Lagerlöfs Vej 300, DK-9220 Aalborg-East, Denmark

² Istanbul Technical University, Science and Technology Institution Building
34469 Maslak, Istanbul

Abstract. With the growing amount of information available on the Web, the task of retrieving documents of interest has become increasingly difficult. Personalized search has got significant attention because it considers the user's preference into the search process to improve the results. However, studies have shown that users are reluctant to provide any explicit input on their personal preference. In this study, we investigate how a search engine can elicit users' preferences by exploring the user's tagging activity from various sources. We propose a simple yet flexible model to succinctly represent user preferences based on multiple factors. Our experiments show that users' preferences can be elicited from a multi-factor tagging data and personalized search based on user preference yields significant precision improvements over the existing ranking mechanisms in the literature.

1 Introduction

Users often have different intentions when using search engines [26]. According to [18], queries provided by users to search engines are under-specified and, thus, poorly translate fully the meaning users have in mind. In fact, the same query provided by different users can have semantically different intentions. For example, one user searching for “stars” may be interested in famous actors or actresses while another may be looking for astronomical objects. As a consequence, according to [12], only 20% to 45% of the results match a user's intentions. One approach to increase user satisfaction is to personalize search results.

Recently, strategies intended to disambiguate the true intention of a query began to collect and analyze user preferences in order to personalize the search retrievals. The user preferences have been modeled in many ways in the literature [4,26,18], including analysis of explicit data such as user profile and preference models, or the implicit collection of data such as user click history, visited pages log, or tagging activity. All of these are indicators of user preferences utilized by search engines to decide which items in the collection of search results are more or less relevant for a particular individual.

In this work, we analyze user's tagging activity to learn the user preferences and personalize the search results. For this purpose, we consider tags because they represent some sort of affinity between user and resource. By tagging, users label resources freely and subjectively, based on their sense of value. Tags, therefore, become a potential source for learning user's interests [8].

In our approach, we build our user model incorporating various tag indicators of user's preference, i.e., each indicator relates to a factor for personalizing searches. We therefore formalize the term *factor* as an indicator of user's preference denoted by a particular set of tags. For instance, a factor Z may represent the set of tags assigned to the most visited pages of a user and a factor Y may represent the set of tags assigned to the pages marked as favorites by the same user. Our belief is that a multi-factor approach can produce a more accurate user model, and thereby facilitate the search for what is more suitable for a given user. The contributions of this paper are:

1. We provide a personalized component to investigate the problem of learning a user's preference based on his tagging activity. We propose a simple yet flexible model to succinctly represent user preferences based on multiple factors.
2. Based on the formal user model, we develop a method to automatically estimate the user's implicit preferences based on his and the community's tagging activity. We provide theoretical and experimental justification for our method.
3. We describe a ranking mechanism that considers a user's implicit interest in ranking pages for a query, based on the users' tagging activity.
4. Finally, we conduct an experimental and comparative analysis to evaluate how much the search quality improves through this personalization. Our experiments indicate reasonable improvement in search quality — we observed about **61.6%** of precision improvement over traditional text-based information retrieval and **6.13%** of precision gain over the best method compared — demonstrating the potential of our approach in personalizing search.

The rest of this paper is organized as follows. We first provide an overview of personalized search and social tagging activity in section 2, on which our work is mainly based. In section 3 we describe our models and methods. Experimental evaluation, results and discussion are presented in section 4. Section 5 addresses the efficiency of the approach and presents the results of retrieval latency. In section 7, we review the related work and finally conclude the paper in section 8.

2 Background

In this section we briefly revise traditional text-based information retrieval concepts followed by an overview of collaborative tagging systems.

2.1 Information Retrieval and Personalized Search

Traditionally, search engines have been the primary tool for information retrieval. According to [2], they appeared to supply the need of human capabilities in storage and retrieval. In brief, they operate by maintaining a comprehensive index of available information. Typically an index represents an information item (such as a Web page or a document) in terms of a set of relevant index terms. When a user submits a query to a search engine, the query terms are compared against index terms for relevant items in the search engine index. An overall relevancy score for each information item (i.e.

document) is computed according to how many of the user's query terms occur in the index, and how important these terms are for that item. The relevant items are then ranked according to their relevancy score before being presented to the user. To produce a composite weight for each term in each document, usually the search score is computed in terms of *term frequency* and *inverse document frequency* (i.e. tf-idf) [2]. The tf-idf weighting scheme assigns to term t a weight in document d given by:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t \quad (1)$$

In other words, $\text{tf-idf}_{t,d}$ assigns to term t a weight in document d that is: i) highest when t occurs many times within a small number of documents (thus lending high discriminating power to those documents); ii) lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal); iii) lowest when the term occurs in virtually all documents.

Eventually the search score of a document d is the sum, over all query terms q , of the tf-idf weights given for each term in d .

$$\text{Score}(q, d) = \sum_{t \in q} \text{tf-idf}_{t,d} \quad (2)$$

In order to achieve personalization, a further step consists of selecting which documents d are relevant or not for a given user. Technically, the traditional scoring function must be adapted/integrated with a personalization model capable of modeling the users' preferences. This is usually done by tracking and aggregating users' interaction with the system. In general, such aggregation includes users' previous queries [22], click-through analysis [13], tagging activity analysis [26,16], etc. A user profile information is also usually employed through personalized query expansion, i.e., adding new terms to the query and re-weighting the original query terms based on the user profile. Similar to [26,16], we explore user's tagging activity as means of learning user's preferences for personalizing search results.

2.2 Collaborative Social Tagging

Collaborative tagging systems have become increasingly popular for sharing and organizing Web resources, leading to a huge amount of user generated metadata. Tags in social bookmarking systems such as *Del.icio.us*¹ are usually assigned to organize and share resources on the Web so that users can be reminded later on. Invariably, tags represent some sort of affinity between user and a resource on the Web. By tagging, users label resources on the Internet freely and subjectively, based on their sense of values. Tags then become a potential source for learning user's interests.

Tagging Notation. Formally, tagging systems can be represented as hypergraphs where the set of vertices is partitioned into sets: $U = \{u_1, \dots, u_k\}$, $R = \{r_1, \dots, r_m\}$, and $T = \{t_1, \dots, t_n\}$, where U , R , and T correspond to users, resources, and tags. A tag annotation, i.e. a resource tagged by a user, is an element of set A , where: $A \subseteq U \times R \times T$. The final

¹ <http://delicious.com>

hypergraph formed by a tagging system is defined as G with: $G = \langle V, E \rangle$ with vertices $V = U \cup R \cup T$, and edges $E = \{\{u, r, t\} \mid (u, r, t) \in A\}$.

Particularly to understand the interests of a single user, our models concentrate on the tags and resources that are associated with this particular user, i.e. in a *personalized* part of the hypergraph G . We then define the set of interests of a user as $I_u = (T_u, R_u, A_u)$, where A_u is the set of tag annotations of the user: $A_u = \{(t, r) \mid (u, t, r) \in A\}$, T_u is the user's set of tags: $T_u = \{t \mid (t, r) \in A_u\}$, R_u is the set of resources: $R_u = \{r \mid (t, r) \in A_u\}$.

The introduction on information retrieval as well as on tagging notations will serve as basis for describing our personalized search model in the following sections.

3 Multi-factor Tag-Based Personalized Search

We now discuss how we personalize search results. In Section 3.1 we first describe our multi-factor representation of user preferences. Then in Section 3.2, we describe how to use this preference information in ranking search results.

3.1 User Preference Representation

Among the numerous amount of resources available on the Web and their diverse subject areas, it is reasonable to assume that an average Web user is interested in a limited subset of articles. Because of this, his activities on the Web such as visiting a page, tagging an item, or bookmarking an interesting Web site are often affected by his preferences. Given these observations, it is convenient to represent the user's preference as a multi-factor user model rather than rely on a single indicator of preference. In this work, tags from different factors such as tags assigned to the user's bookmarks and his own tags serve as our learn units of user preferences. Further, we understand that some tags are preferred over another, meaning that the frequency of usage of a given tag can denote its affinity with the tagger. In this sense, we define the tag preference set, for each factor, as a tuple $(tag, tagFreq(tag))$, where $tagFreq(tag)$ is a function that measures the user's degree of interest in that tag. Formally, we define this set for a particular user u and factor $f \in F$ (let F be the set of all possible factors) as:

$$T_f = \{(t, tagFreq(t)) \mid t \in T_f\}, \quad (3)$$

where $tagFreq(t) = \frac{n_t}{|T_f|}$, n_t is the number of occurrences of the tag $t \in T_f$ and $|T_f|$ is the amount of tags in a given factor f . The set T_f is normalized such that $\sum_{i=1}^{|T_f|} tagFreq(i) = 1$. To illustrate the user tagging preference representation, suppose a user has only two tags in one particular factor: "semantic web" and "data mining", and the first has been utilized three times while the second has been utilized only once. This means the user has been interested in "semantic web" three times as much as he has been interested in "data mining". Then, the tag preference set of the user for that factor will be represented as $\{("semantic web", 0.75), ("data mining", 0.25)\}$.

The composition of our multi-factor tag-based user model T'_u extends the traditional set of user tags T_u (see subsection 2.2), with a disjoint union of tag sets:

$$T'_u = \bigsqcup_{f \in F} T_f, \quad (4)$$

where T_f is the set of tags assigned to each factor $f \in F$. Next section explains how the tag-based multi-factor approach is applied to personalize search results.

3.2 Tag-Based Personalization Approach

The tag-based personalization approach decides which resource $r \in R$ is relevant to each user $u \in U$ based on his preferences established in the multi-factor tag-based user model. In the context of this work, we re-rank the search results by measuring the similarity of tags that denote user preference and the tags assigned to the retrieved items. With this, we promote the items closer to user's preferences at the first positions in the collection of search results.

Technically, we calculate the cosine similarity between *each* vector of tag frequencies $\vec{T}_f \subset \vec{T}'_u$ from $T_f \subset T'_u$ and the vector of tag frequencies \vec{T}_r from the retrieved resources T_r of a given user query q . Further, we weigh each vector \vec{T}_f with a coefficient, α_f that determines its degree of importance over the other factors in the model. We incorporate the coefficients because different users may rely on tag factors differently. As a consequence, the importance of each factor may vary accordingly.

The coefficient values for each factor are automatically estimated using the Ordinary Least Square (OLS) linear regression. The goal of using OLS is to minimize the sum of squared distances between the observed tag frequencies of each factor in the dataset and the best coefficient values predicted by linear approximation [25].

Once the coefficient values are estimated, the Tag-based Similarity Score (TSS) can be calculated as:

$$TSS(\vec{T}_r, \vec{T}'_u) = \sum_{f=1}^{|F|} \alpha_f * \frac{\vec{T}_r \cdot \vec{T}_f}{|\vec{T}_r| |\vec{T}_f|}, \quad (5)$$

The TSS value is then utilized to weigh the ordinary search score and thereby promote the items matching the user interests to higher positions in the search ranking. The Personalized Search Score (PSS) over a resource r triggered by a query q in the set of resources R is then defined as follows:

$$PSS(q, r, u) = \sum_{t \in q} \text{tf-idf}_{t,r} * TSS(\vec{T}_r, \vec{T}'_u). \quad (6)$$

In the following section, with the intuit of demonstrate the how the personalized search takes place, we simulate a user search since the user's initial query until the re-ranking of search results.

3.3 Personalized Search in Action

In this section we describe how the personalized ranking is realized when a user $u \in U$ submits a query q for a search engine:

1. Assume that a user u , whose preferences are denoted by the multi-factor tag-based model T'_u , submits a query q for a given search engine;

2. The search engine returns a set of resources $S \subseteq R$ that matches the entry query q . Each retrieved resource $s \in S$ is assigned with a set of tags defined by $T_s = \{t_1, \dots, t_z\}$. The retrieved items in S are initially ranked respecting the tf-idf ordering $\tau = [s_1, s_2, \dots, s_k]$, where the ordering relation is defined by $s_i \geq s_j \Leftrightarrow \text{tf-idf}(q, s_i) \geq \text{tf-idf}(q, s_j)$.
3. For each $s \in S$, we weigh the τ with $TSS(\vec{T}_s, \vec{T}'_u)$. The outcome is a personalized ranking list of pages represented by S' . The rank list will follow a new ordering $\tau' = [s_1, s_2, \dots, s_k]$, where $s_i \in S'$ and the ordering relation is defined by $s_i \geq s_j \Leftrightarrow PSS(q, s_i, u) \geq persScore(q, s_j, u)$.
4. The personalized result set $S' \subseteq R$ is then returned to the user.

4 Experimental Evaluation

To evaluate our approach, we analyze and compare how the different similarity measures re-rank the returned result list. For the matter of comparison, the best personalization approaches will rank the relevant/preferable retrieved items to the higher positions in the result list.

We utilize the precision metric to measure and compare the performance of our personalization approach over other similarity measures. By doing so, the research goal of this evaluation is *to assess whether the multi-factor tags improve the precision of traditional text-based information retrieval mechanism in leading users to find the information they are actually looking for*. We intend to address (and attenuate) the problem witnessed by searchers when the search results do not meet their real needs and, instead, present them with undesired information.

4.1 MovieLens Dataset

We have utilized the MovieLens dataset for evaluating our approach. The data set contains 1,147,048 ratings and 95,580 tags applied to 10,681 movies by 13,190 users. The dataset was chosen because it allowed us to build multi-factor user model base on two distinct factors corresponding to two different user activities: tagging and rating. The first factor refers to T_u as the set of user's tags and the second refers to V_u , the set of tags assigned by other users to the pages rated four or five stars by user u . In a scale from one to five, we consider movies rated four or five as strong indicators of user's preferences. The evaluated multi-factor user model is defined as follows:

$$T'_u = \gamma.T_u + \beta.V_u \quad (7)$$

where γ and β are the coefficients used to weigh the importance of each tag factor.

Because the MovieLens dataset only provides little information about the movies such as title and category, we crawled the synopses of the movies in order to create our search space. The synopses were extracted from *The Internet Movie Data Base* (IMDb)², a movie repository that catalogs information about movies, including information about casting, locations, dates, synopses, reviews, and fan sites on the Web.

² <http://www.imdb.com/>

4.2 Methodology and Evaluation Metrics

In order to evaluate whether the proposed approach matches our research goal, we performed a quantitative evaluation in terms of precision of the search results. Precision refers to the proportion of retrieved items that are actually relevant to the user's query. We calculated the precision as $prec(q, u) = \frac{|R_{q,u}| \cap |R'_{q,u}|}{|R_{q,u}|}$, where $|R_{q,u}|$ is the amount of retrieved items for a query q triggered by an user u and $|R'_{q,u}|$ is the amount of *relevant* pages expected to be retrieved. This set was composed by movies rated four or five stars besides the query q as a term in the text content. In this way, we could distinguish what was relevant or not for each user. Further, we statistically computed the most representative number of users to be assessed. The sample size was calculated with confidence level set to 95% and confidence interval set to 2%.

Queries and Search Space. For each user, we ran the top n queries using the most popular terms that appeared in all indexed pages. We prioritized the most frequent terms along all documents and the most frequent ones in each document. It is important to mention that we filtered out the stop words prior to processing the list of indexed terms. This list was calculated as a mean of tf-df [2]. This decision was taken to avoid that any particular group of individuals was favored over another.

Once we have selected the queries, we focused on defining the most appropriate search space. Since our goal was centered on the precision, we decided to focus the search only at the top 30 ranked items. This focused observation was motivated by the fact that users usually don't go further to encounter what they are looking for, instead they reformulate queries to better convey the information they are actually seeking [2].

Pruning. Although substantial data was available, we pruned the dataset for our analyses of the personalized search. We focused on a set of movies with a minimum threshold of tags and ratings. We began pruning the present dataset by selecting items that had been tagged with at least 5 distinct tags. We wanted to focus on tag sets with high lexical variability and less redundancy. This care was taken to prevent that tags from distinct factors could overlap with tags from expected documents in the case of low tag variability. We iteratively repeated this pruning until reach a stable set of items and tags. Further, since we wanted to explore the precision of the personalized search, we only included users that had distributed ratings higher than or equal to four stars. Otherwise we would not have any source of information on user's preference about the retrieved items. The complete overview of the dataset size is shown in Table 1.

Similarity Measure Comparison. The core of the tag-based personalization approach is the computation of the Tag-based Similarity Score (TSS) (see in section 3.2). In order to compare our approach with other similarity measures, we changed the TSS algorithm using other similarity algorithms without considering the coefficients assigned to the factors involved. Besides the cosine similarity, the other similarity measures utilized are the *Matching Coefficient*, *Dice*, *Jaccard* and *Euclidean Distance* (all refer to [5]).

Table 1. Size of different datasets we utilized in this paper. *Count* is the number of movies the dataset contains. *N. users* is the number of users that generated those entities. For example, the second and third columns of the second row indicate that 4,009 users assigned 95,580 tags to movies. Total is the number of times the users rated or tagged the movies. The last three columns indicate the same numbers after the pruning is being applied.

Dataset	Before Pruning			After Pruning		
	Count	N. Users	Total	Count	N. Users	Total
Movies rated	8,358	9,753	1,147,048	7,075	6,476	211,746
Movies tagged	7,601	4,009	95,580	4,029	3,871	88,181

- The *Matching Coefficient* approach is a simple vector based approach which simply counts the number of terms (tags in our case), (dimensions), on which both vectors are non zero. So, for vector set \vec{v} and set \vec{w} the matching coefficient is calculated as $matching(\vec{v}, \vec{w}) = |\vec{v} \cap \vec{w}|$. This can be seen as the vector based count of similar tags.
- The *Dice Coefficient* is a term based similarity measure (0-1) whereby the similarity measure is defined as twice the number of terms common to compared entities divided by the total number of tags assigned in both tested resources. The Coefficient result of 1 indicates identical vectors (e.g. \vec{v} and \vec{w}) as where a 0 equals orthogonal vectors. The coefficient can be calculated as $dice(\vec{v}, \vec{w}) = \frac{2|\vec{v} \cap \vec{w}|}{|\vec{v}| + |\vec{w}|}$.
- The *Jaccard Coefficient* measures similarity between sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. The coefficient can be calculated as: $jaccard(\vec{v}, \vec{w}) = \frac{|\vec{v} \cap \vec{w}|}{|\vec{v} \cup \vec{w}|}$.
- The *Euclidean Distance* approach works in vector space similar to the matching coefficient and the dice coefficient, however the similarity measure is not judged from the angle as in cosine rule but rather the direct euclidean distance between the vector inputs. The standard Euclidean distance formula between vectors \vec{v} and \vec{w} is defined as follows: $euclidean(\vec{v}, \vec{w}) = \sqrt{\sum_{i=1}^n (\vec{v}_i - \vec{w}_i)^2}$.
- The *Cosine Similarity* is utilized to measure the similarity between two vectors of n dimensions by finding the cosine of the angle between them. The cosine similarity for two vector between vectors is calculated as: $cosSim(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|}$.

Unlike our proposed TSS algorithm, here we do not consider the coefficients assigned to the tag factors. For the sake of differentiation, we call the cosine similarity with coefficient applied to our TSS algorithm as $Cosine^{Coef}$.

4.3 Evaluation Results

Figure 1 shows the mean of precision (Subfigure 1(a)) and the box plot displays the distribution differences between the five similarity measures (Subfigure 1(b)). The higher precision values correspond to better performance. The precision of our approach is displayed in the first bar ($Cosine^{Coef}$) followed by the compared similarity measures with

Table 2. Mean of Precision

	Cosine ^{Coef}	Cosine	Jaccard	Dice	Euclidean	Matching
Min.	0.723	0.616	0.576	0.575	0.600	0.510
1st Qu.	0.765	0.710	0.576	0.577	0.600	0.583
Median	0.776	0.761	0.634	0.633	0.634	0.583
Mean	0.796	0.755	0.694	0.694	0.667	0.641
3rd Qu.	0.841	0.845	0.841	0.842	0.756	0.766
Max.	0.912	0.909	0.912	0.913	0.886	0.865
Std.Dev	0.052	0.087	0.131	0.130	0.083	0.105

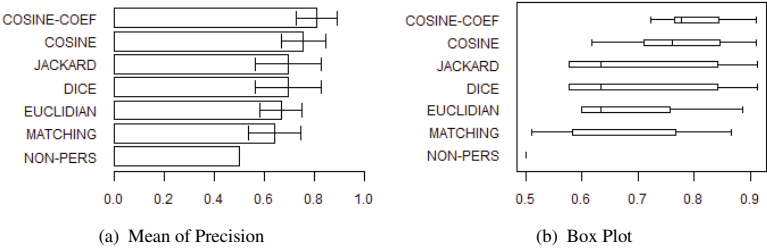


Fig. 1. Overall result and statistics from the experiment. Note that each of the similarity measure are utilized in the TSS algorithm to personalize the search.

their respective standard errors. We also include the non-personalized precision result for the sake of comparison. This ranking was based on the traditional tf-idf weighting scheme where the text-based search has no support from personal tags. It is important to emphasize that all compared similarity measures produce a personalized score except the non-personalized search.

Table 2 summarizes the statistics from our sample of precision values regarding the evaluated metrics. It shows the five-number summary (sample minimum, lower quartile, median, upper quartile and sample maximum), the mean value, and the standard deviation of the samples. It is worth noting that the sample minimum value of our approach has higher precision than any other similarity measure compared. This behavior is also noticed when compared with the sample maximum value. Similarly, the standard deviation of the sample of our approach is lower than the standard deviation of the other samples. This indicates less dispersion within the precision values of our approach. Furthermore, differences between all pairs of similarity measures are significant ($p \leq 0.01$) except for those between Jaccard and Dice similarity.

As results show, our approach (Cosine^{Coef}) achieved the highest precision rates. In particular, it achieved 61.6% of precision improvement over traditional text-based information retrieval (non-pers) and 6,13% of precision gain over cosine similarity. The overall result also indicates that the cosine-based similarity measures (regarding or not the coefficients) perform better than the other approaches. As expected, all similarity measures applied to the personalized search outperformed the non-personalized method.

5 Assessment of Efficiency

The efficiency of information retrieval systems is of utmost importance, because it ensures that systems scale up to the vast amounts of information needing retrieval. Efficiency research over the past years has focused on efficient indexing, storage (compression) and retrieval of data (query processing strategies). In the previous section we explored the comparative evaluation of the quality of the search results. Although, our approach shows significant improvement over the compared methods, there is still always a cost of performance involved. In this Section, we aim at demonstrating the impact of the personalization on the retrieval latency as well as the cost for building the personalized ranking.

To quantify the efficiency of our approach, we run a large number of keyword queries on an Intel Core 2 Duo E8400 with 8GB Ram running Sun Java 6 on a 32-bit installation of Windows XP. The structure of the data as well as the indexed terms are stored in a PostgreSQL object-relational database. The algorithm is not parallelized and runs entirely on a single core. We do not use any partitioning or segmentation techniques, but hold all individual searches in memory, using about 2 GB of main memory.

5.1 Indexing Time and Personalization Cost

The total indexing time of the all 211,746 test documents (rated and tagged) was 135 seconds. Within this time, the invalid characters are steamed out the term frequency are calculated. To conduct this task, we utilized the Apache Lucene search engine ³. We consider this time relatively feasible in case there is a need to re-index the whole corpus in a real time application.

Invariably, the calculation of the user preferences adds additional cost to the overall search. The cost is related to the computation of tag frequencies along the tag factors. Regardless this work, personalization does not take place. We therefore measured the cost of personalization for 500 users randomly chosen from our dataset. We ran ten queries for each user and the response time is given as the mean of them. As a result, we evidenced that, on average, the calculus of user preferences takes approximately 910 ms. This value is latter added to the overall retrieval.

5.2 Data Retrieval Latency

In order to evaluate the retrieval latency, we analyzed how the latency scales according to different scenarios setups. The scenarios are assembled in terms of number of documents, number of tags and number of users. We then collected the latency for the non-personalized method and our approach (called Cosine^{Coef}). In this evaluation, the users' preferences are calculated on demand. The other similarity approaches were not considered in this evaluation since the difference of retrieval latency are not significant and the cost of personalization is approximately the same for all methods. For each test scenario, 10 queries were issued per user. These queries however are not the same as the previous assessment on the personalization cost.

³ <http://lucene.apache.org/>

Table 3. Data Retrieval Latency Scenarios

Scenario ID	N. Documents	N. Tags	N. Users	Non-Personalized (ms)	Cosine ^{Coef} (ms)
1	500	1500	100	380	1347
2	1000	5000	500	664	1452
3	5000	10000	1000	844	1687

According to Table 3, it is possible to observe that the retrieval latency is relatively low affected as the scenarios scale up. Unsurprisingly, the non-personalized search outperforms our approach in all test scenarios. On the first scenario, the efficiency of the non-personalized method achieves, on average, 380 ms whereas the personalized one achieves 1347 ms. In comparison with the previous assessment on the personalization cost (see subsection 5.1), we notice that the current findings are relatively close to previous result since the cost is 967 ms. The slight difference from previous result (from 967 ms to 910 ms) concerns basically the fact that different queries might have been issued. As a consequence, the result set of some queries might have been bigger than the others, thus impacting in the whole retrieval latency.

The difference between the latency of non-personalized method and our approach reduces as the scenarios scale down. On the first scenario, our approach is 71.7% more expensive than the non-personalized method. On the second scenario, such a difference reduces to 54.02% and finally on the third scenario, the difference reduces to 49.9%. Interestingly, this observation demonstrates that the computation of users' preferences is fairly affected as the amount of users and tags evolve.

Another interesting finding can be made on the retrieval latency of each method individually. From the first to the second scenario, the non-personalized approach decreases its efficiency in 42.8% and from the second to the third scenario, the same method decreases its efficiency in 21.4%. This behavior is inverse when comes to the personalized approach. From the first to the second scenario, our approach decreases its efficiency in 7.2% and from the second to the third scenario, it decreases its efficiency in 13.9%. In comparison with the non-personalized method, we can observe that our approach is relatively more efficient as the scenario scales up.

6 Discussion and Limitations

Concerning the role of tags as means of personalizing traditional information retrieval, we evidenced that tags indeed can be used as learning units of user's preference. As results showed, this hypothesis was confirmed when the precision of our solution considerably outperformed the precision of non-personalized search. The immediate benefit of personalized search is the reduction of undesired retrievals at the first positions and thereby not distracting users with unsolicited information. Part of this success relates to the best estimation of coefficients applied to each factor considered in the search. The adoption of coefficients helped us to efficiently determine which tag factors are more representative for an individual over the others. Regarding the applicability of the approach, we are quite positive that existing tag-based systems can utilize and/or adapt our solution to their reality.

The main limitation of the current approach is the lack of tags. In this work we are not addressing this problem since the goal is to emphasize the benefits with the multi-factor approach. A potential solution is seen in the work of [12] that studies tag prediction in social bookmarking environments. Performance is another issue that was not formally evaluated in the current study but we empirically observed that the personalization process is at least 10% more expensive than the basic keyword-based search. However this cost cannot be judged isolated since the personalized outcome can compensate the investment. Further investigation is necessary and planned for future works.

7 Related Work

In recent years many researchers utilize query log and click-through analysis for web search personalization. In [18], the authors combine a topic-sensitive version of PageRank [11] with the history of user clicks data for search result personalization. Joachims et al. [13] study clicks applicability as implicit relevance judgments. They show that users' clicks provide a reasonably accurate evidence of their preferences. [21] explore the use of a less-invasive means of gathering user information for personalized search. In particular, they build user profiles based on activity at the search site itself and study the use of these profiles to provide personalized search results. These studies are relevant to our work in context of predicting user preferences but our approach takes into account user annotations which are called tags as user feedback. Tan et al. [22] propose a language modeling approach for query history mining. Their small-scale study demonstrates significant improvement of personalized web search with a history-based language model over regular search. The user modeling approach described in [19] is based on a decision-theoretic framework to convert implicit feedback into a user profile that is used to re-rank search results.

While user models are usually targeted at search personalization, they could also be applied for personalized information filtering, as was shown in [27] which analyzes click history for the identification of regular users' interests. The work from [24] shows that combining implicit user profiles from several related users has a positive impact on personalization effectiveness. Recently, new approaches for adaptive personalization focus on the user task and the current activity context. There are several approaches trying to predict applicability of personalization while considering the current context of the user's task on query submission [23,7].

Collaborative Filtering (CF) [3] has become a very common technique for providing personalized recommendations, suggesting items based on the similarity between users' preferences. One drawback of traditional CF systems is the need for explicit user feedback, usually in the form of rating a set of items, which might increase users' entry cost to the recommender system. Hence, leveraging implicit user feedback [1], such as views, clicks, or queries, has become more popular in recent CF systems. In this work, we leverage implicit tagging information, which can be viewed as a variant of implicit user feedback. [6] proposes a solution which considers the structure of the user's social network assuming that people which share same social ties have similar interests. In addition to social network, they have another approach similar to our study taking into account user tags to understand user interest. Comparing these two approaches they

noted that tagging activity gives efficient information about user preferences for active taggers.

In order to measure relevance between user and search result, [10] considers topic matching instead of tags which we used in our approach. They propose a ranking algorithm which ranks web pages by the term matching between user interest and resource's topic. [9] proposes a (meta-)search engine which queries more than 18 commodity search engines and offers two complementary views on their returned results. One is the classical flat-ranked list, the other consists of a hierarchical organization of these results into folders created on-the-fly at query time and labeled with intelligible sentences that capture the themes of the results contained in them. Users can browse this hierarchy with various goals: knowledge extraction, query refinement and personalization of search results. SnakeT personalizes on-the-fly the original ranked list by filtering out those results that do not belong to the selected folders by users. In [20], there exists a different approach to personalizing search result by building models of user as ontological profiles. They derive implicit interest scores to existing concepts in domain ontology. A different way of personalized search is based on global interests using the Open Directory Project (ODP) categories [14]. In [15], a ranking algorithm is defined similar to previous work using ODP categories for categorizing and personalizing search result. [14] explore ways of incorporating users' interests into the search process to improve the results using an domain ontology. The user profiles are structured as a concept hierarchy of 4,400 ontology classes. The disadvantage of structured user profiles is the need of constant revision of such models and the cost of maintenance.

Focused mostly on tags, [26] proposes an algorithm for personalizing search and navigation based on personalized tag clusters. Slightly similar to our model, they measure the relevance between user and tag cluster and try to understand user interest while we calculate the similarity between user and tags. Tags are used in different manners to find search personalized results answering user needs since tags are chosen by users personally. Similar to our study, in [16], a tag based re-ranking model is presented taking into account tags from *Del.icio.us*. Likewise our model, they compare tags of search results and users to calculate new scores of search results.

8 Conclusions and Future Works

This study introduces a multi-factor tag-based model to personalize search results. We analyze user's tagging activity to learn users' preferences and use this information to personalize the search. We evaluated our approach with other personalization methods and as a result we realized significant improvement of precision. As a future work, we intend to analyze the semantic relationship between tags in order to catch hidden similarities that are not undertaken by this model. In addition, we aim at enhancing the model by considering the tag decay. The goal is to perform a temporal analyzes and filter the results according to the actual users' preferences.

Acknowledgements. The research leading to these results is part of the project “KiWi - Knowledge in a Wiki” and has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211932.

This work has been supported by FP7 ICT project M-Eco: Medical Ecosystem Personalized Event-Based Surveillance under grant number 247829. The authors would like to acknowledge GroupLens research at University of Minnesota for providing the MovieLens dataset.

References

1. Au Yeung, C.M., Gibbins, N., Shadbolt, N.: A study of user profile generation from folksonomies. In: *Proceedings of the Workshop on Social Web and Knowledge Management (SWKM 2008)*, Beijing, China, April 21–25, 2007, pp. 1–8 (2008)
2. Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. Addison Wesley (1999)
3. Bender, M., Crecelius, T., Kacimi, M., Michel, S., Neumann, T., Parreira, J.X., Schenkel, R., Weikum, G.: Exploiting social relations for query expansion and result ranking. In: *ICDE Workshops*, pp. 501–506. IEEE Computer Society (2008)
4. Biancalana, C.: Social tagging for personalized web search. In: *AI*IA 2009: Emergent Perspectives in Artificial Intelligence*, pp. 232–242 (2009)
5. Boninsegna, M., Rossi, M.: Similarity measures in computer vision. *Pattern Recognition Letters* 15(12), 1255–1260 (1994)
6. Carmel, D., Zwerdling, N., Guy, I., Ofek-Koifman, S., Har'el, N., Ronen, I., Uziel, E., Yegorov, S., Chernov, S.: Personalized social search based on the user's social network. In: *CIKM 2009: Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pp. 1227–1236. ACM, New York (2009)
7. Dou, Z., Song, R., Wen, J.-R.: A large-scale evaluation and analysis of personalized search strategies. In: *WWW 2007: Proceedings of the 16th International Conference on World Wide Web*, pp. 581–590. ACM, New York (2007)
8. Durao, F., Dolog, P.: A personalized Tag-Based recommendation in social web systems. In: *Proceedings of International Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0 2009)* at UMAP 2009, vol. 485 (2009)
9. Ferragina, P., Gulli, A.: A personalized search engine based on web-snippet hierarchical clustering. *Software: Practice and Experience* 38(2), 189–225 (2008)
10. Gemmell, J., Shepitsen, A., Mobasher, M., Burke, R.: Personalization in folksonomies based on tag clustering. In: *Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems* (2008)
11. Haveliwala, T.H.: Topic-sensitive pagerank. In: *WWW 2002: Proceedings of the 11th International Conference on World Wide Web*, pp. 517–526. ACM, New York (2002)
12. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies, pp. 506–514 (2007)
13. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting click-through data as implicit feedback. In: *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 154–161. ACM, New York (2005)
14. Liu, F., Yu, C., Member, S., Meng, W.: Personalized web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering* 16, 28–40 (2004)
15. Ma, Z., Pant, G., Sheng, O.R.L.: Interest-based personalized search. *ACM Trans. Inf. Syst.* 25(1), 5 (2007)
16. Noll, M.G., Meinel, C.: Web Search Personalization Via Social Bookmarking and Tagging. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) *ASWC 2007 and ISWC 2007*. LNCS, vol. 4825, pp. 367–380. Springer, Heidelberg (2007)

17. Pretschner, A., Gauch, S.: Ontology based personalized search. In: 1999 Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence, pp. 391–398 (1999)
18. Qiu, F., Cho, J.: Automatic identification of user interest for personalized search. In: WWW 2006: Proceedings of the 15th International Conference on World Wide Web, pp. 727–736. ACM, New York (2006)
19. Shen, X., Tan, B., Zhai, C.: Implicit user modeling for personalized search. In: CIKM 2005: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 824–831. ACM, New York (2005)
20. Sieg, A., Mobasher, B., Burke, R.: Web search personalization with ontological user profiles. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 525–534. ACM, New York (2007)
21. Speretta, M., Gauch, S.: Personalized search based on user search histories. In: IEEE / WIC / ACM International Conference on Web Intelligence, pp. 622–628 (2005)
22. Tan, B., Shen, X., Zhai, C.: Mining long-term search history to improve search accuracy. In: KDD 2006: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 718–723. ACM, New York (2006)
23. Teevan, J., Dumais, S.T., Liebling, D.J.: To personalize or not to personalize: modeling queries with variation in user intent. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 163–170. ACM, New York (2008)
24. Teevan, J., Morris, M.R., Bush, S.: Discovering and using groups to improve personalized search. In: WSDM 2009: Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 15–24. ACM, New York (2009)
25. Williams, G.: Linear Algebra With Applications. Jones and Bartlett Publishers, Inc., USA (2007)
26. Xu, S., Bao, S., Fei, B., Su, Z., Yu, Y.: Exploring folksonomy for personalized search. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 155–162. ACM, New York (2008)
27. Yang, B., Jeh, G.: Retroactive answering of search queries. In: WWW 2006: Proceedings of the 15th International Conference on World Wide Web, pp. 457–466. ACM, New York (2006)