Optimizing single term queries using a personalized Markov random walk over the social graph

Maarten Clements[1,2], Arjen P. de Vries[2,1], Marcel J.T. Reinders[1]

[1]Information and Communication Theory Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands. [2]National Research Institute for Mathematics and Computer Science (CWI), The Netherlands.
E-mail: {m.clements, m.j.t.reinders}@tudelft.nl, Arjen@acm.nl

ABSTRACT

Social content systems contain enormous collections of unstructured user-generated content, annotated by the collaborative effort of regular Internet users. Tag-clouds have become popular interfaces that allow users to query the database by clicking relevant terms. However, these single click queries are often not expressive enough to effectively retrieve the desired content.

Using both rating and tagging information we have created a personalized retrieval model that effectively integrates the personal user preference in the content ranking. The soft clustering effect of our random walk model allows a smooth integration of concepts indirectly related to the target user and the query tag.

With collaborative annotations from a popular on-line book catalog, we show that our model outperforms standard tag-based retrieval. Both personalization and smoothing with closely related concepts significantly improve the content ranking. Our results indicate that individually created annotations are not semantically expressive enough to enable effective retrieval. Finally, we discuss the robustness of our model to well known linguistic problems like synonyms and homographs.

Keywords

Social Networks, Content Retrieval, Collaborative Tagging, Rating, Personalization

1. INTRODUCTION

In the last decade, the explosive use of digital budget cameras and integrated multimedia devices has resulted in an enormous increase in user-generated multimedia content like movieclips and pictures. On-line databases are actively used to store and share this content. Recently, the addition of social aspects in these databases has resulted in a large popularity increase. Millions of people use these *social content systems* to publish their creations or to be entertained by other people's contributions. Because the contributed data often does not carry a clear contextual description and there is no librarian to categorize the content, this has resulted in huge collections of unstructured data.

For future retrieval, many network users actively annotate the content using tags. Although most people use tagging to organize their own content collection, it has been shown that social tagging results in semantically descriptive annotations that can be used for content retrieval by the entire network [5, 11]. To initiate content retrieval, social tags are often shown in a *tag-cloud*, a visual depiction of tags in which the more popular tags are typeset in a larger font or more prominent color. Although there exist many different methods to draw these clouds [9], the relevance of a tag is often based on the global popularity of the tags in the entire network (e.g. popular tags in Last.fm[1]). In this way of navigation only a single popular word is used as a query, resulting in many retrieved documents. In traditional information retrieval (web-search engines), people often use multiple word queries in order to disambiguate their information need. To enable effective content ranking based on a single term, social content systems should be personalized to the user's preference.

In currently popular social content systems, there is a difference between *collaborative* tagging systems (e.g. CiteULike[2] and Del.icio.us[3]) and *individual* tagging systems (e.g. YouTube[4] and Flickr[5]). Many systems that allow user-generated content injection are individual tagging systems where only the injector of the content is able to assign the tags. In these systems, many people (who do not contribute any content) will not build up a profile of the tags they prefer. In collaborative tagging (CT), every user can tag any piece of content. In this way, users indicate which aspects of the content correspond to their personal interest. Also, in CT systems the aggregated tags of the network users create a relevance distribution for each content element. Furnas et al. already stated in 1987 that people often choose different terms to annotate content, resulting in low precision retrieval [4]. They argued that a theoretically optimal system would allow *unlimited aliasing* to describe the content. We advocate that collaborative tagging approaches unlimited aliasing and is therefore required to enable effective personalized content retrieval.

Besides tagging, the social aspects of networks stimulate people to share their opinion about the provided content. In many interfaces people can assess the quality of the content by giving a rating. With the introduction of ratings and tags in on-line databases, content annotation has shifted to subjective categorization. The combination of these two information sources creates a non-hierarchical database categorization based on both content quality and topic. Using ratings and tags, we create a graph of the network, resembling the actual relations in social content systems. We use a personalized random walk over this graph to evaluate the retrieval performance of single click queries.

[1]http://www.last.fm/tags
[2]http://www.citeulike.org
[3]http://del.icio.us
[4]http://www.youtube.com
[5]http://www.flickr.com

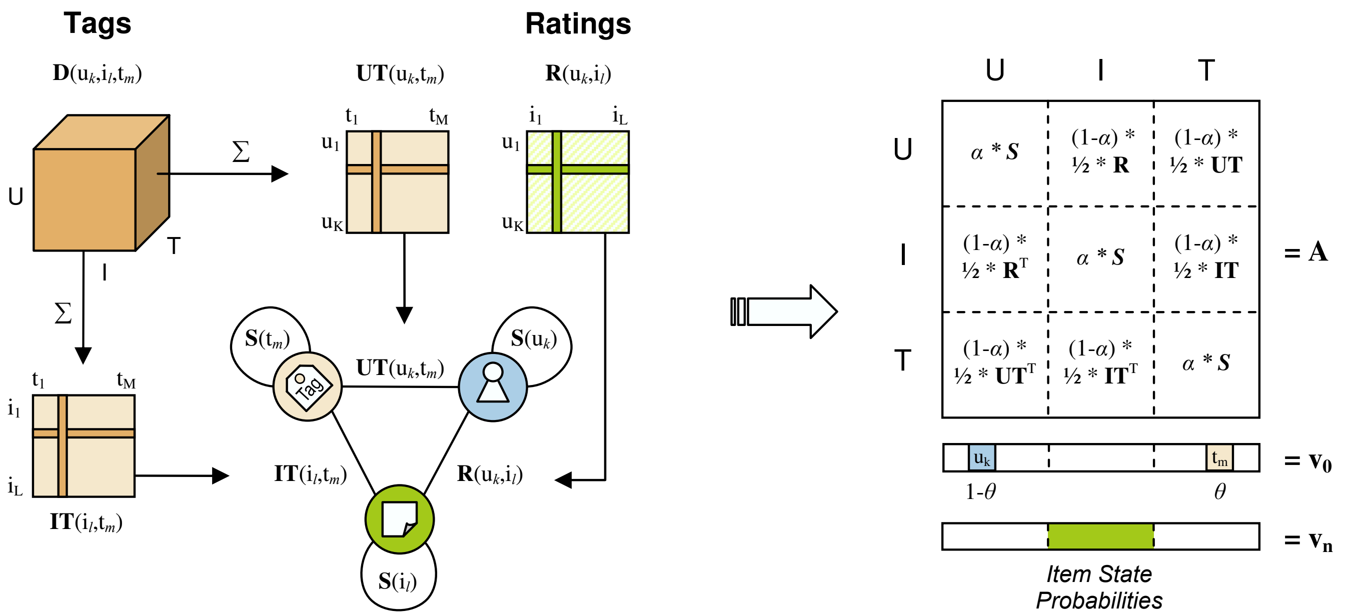

Figure 1: In our random walk model, the social content network is represented as a tripartite graph, containing users, items and tags as nodes. The edges between these entities are determined by rating ($\mathbf{R}$) or tag count ($\mathbf{UT}$, $\mathbf{IT}$). Self transitions ($\mathbf{S}$) allow the random walk to stay in the same node with a certain probability. Together, these edges constitute transition matrix $\mathbf{A}$. In the initial state vector $\mathbf{v}_0$, the index corresponding to the target user and the selected query tag are assigned with weights $1-\theta$ and θ. The result of the walk $\mathbf{v}_n$ contains the relevance probabilities of all three network elements. The model parameter α is used to tune the influence of self transitions.

2. PERSONALIZATION MODEL

For the relevance ranking of the content based on a selected tag we propose to use a random walk over the social graph, created by all rating and tagging actions. A random walk is a stochastic process in which the initial condition is known and the next state is given by a certain probability distribution. This distribution can be represented by the *transition matrix* $\mathbf{A}$, where $\mathbf{A}_{i,j}$ contains the probability of going from node i (at time n) to j (at time $n+1$):

$$\mathbf{A}_{i,j} = P(S_{n+1} = j | S_n = i) \tag{1}$$

The initial state can now be represented as a vector $\mathbf{v}_0$ (with $\sum(\mathbf{v}_0) = 1$), in which the query elements can be assigned. By multiplying the state vector with the transition matrix, we can find the state probabilities after one step in the graph ($\mathbf{v}_1$). Multi step probabilities can be found by repeating the multiplication $\mathbf{v}_{n+1} = \mathbf{v}_n\mathbf{A}$, or using the n-step transition matrix $\mathbf{v}_n = \mathbf{v}_0\mathbf{A}^n$. The number of steps taken in the random walk determines the influence of the initial state vector versus the background distribution. Under certain graph conditions, $\mathbf{v}$ will become stable (so that $\mathbf{v}_\infty = \mathbf{v}_\infty\mathbf{A}$) and in a completely connected graph it will contain the background probability of all nodes in the network (determined by the *volume* of connected paths).

Figure 1 shows how we create the transition matrix by combining rating and tagging information. If users, items and tags are seen as separate entities, the act of tagging creates a ternary relation between them [12]. These relations can be visualized in a 3D matrix $\mathbf{D}(u_k, i_l, t_m)$, where each position indicates if user u_k (with $k = \{1, \ldots, K\}$) tagged item i_l (with $l = \{1, \ldots, L\}$) with tag t_m (with $m = \{1, \ldots, M\}$).

Because even collaborative tagging systems are usually very sparse, we propose not to use the ternary relations directly, but sum over the 3 dimensions of $\mathbf{D}$ to obtain:

UT matrix: $\mathbf{UT}(u_k, t_m) = \sum_{l=1}^{l=L} \mathbf{D}(u_k, i_l, t_m)$, indicating how many items each user tagged with which tag.

IT matrix: $\mathbf{IT}(i_l, t_m) = \sum_{k=1}^{k=K} \mathbf{D}(u_k, i_l, t_m)$, indicating how many users tagged each item with which tag. In individual tagging systems, this will be a binary matrix.

UI matrix: $\mathbf{UI}(u_k, i_l) = \sum_{m=1}^{m=M} \mathbf{D}(u_k, i_l, t_m)$, indicating how many tags each user assigned to each item.

Earlier work on collaborative tagging systems proposed to create the social graph from the three projections of the ternary user-item-tag relation [12, 7]. Although the **UI** matrix contains interesting information about the users' tagging behavior, the relation between the number of tags assigned to an item and the preference of the user toward that item is unclear. Therefore, when modeling the users' preference, we replace the tag based User-Item matrix by the matrix based on the users' ratings. The rating matrix ($\mathbf{R}(u_k, i_l)$) contains the explicit users' preference for the available content, often expressed on a five or ten point scale.

Using the nonzero matrix values as edges, these three matrices ($\mathbf{UT}$, $\mathbf{IT}$ and $\mathbf{R}$) constitute a tripartite graph with users, items and tags as nodes. We include self-transitions that allow the walk to stay in place, which increases the influence of the initial state. The self transitions are represented in a diagonal matrix of ones $\mathbf{S} = \mathrm{diag}(1, \ldots, 1)$, so that the weight of the self transitions is equal for all nodes.

To reduce the influence of frequently occurring elements, we use TF-IDF weighing on the input matrices [15]. For example, the weighted User-Tag matrix is computed by:

$$\mathbf{UT}_{\text{TF-IDF}}(u_k, t_m) = \frac{\mathbf{UT}(u_k, t_m)}{\log \sum_{u=1}^{u=K} \mathrm{sgn}(\mathbf{UT}(u_k, t_m))} \tag{2}$$

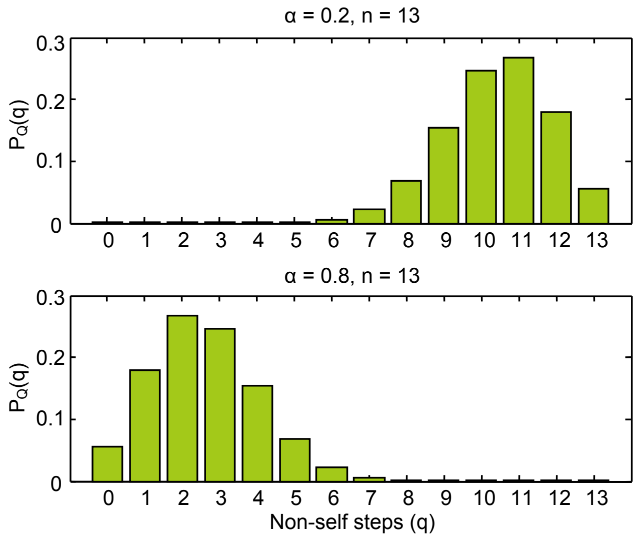

Figure 2: The PMF of the number of non-self steps after 13 steps through the social graph, for $\alpha = 0.2$ and $\alpha = 0.8$. Note that the number of non-self steps does not equal the distance to that starting node, because the walk might revisit earlier passed nodes.

where the sign function (sgn) sets all values > 0 to 1. Before combining the matrices we normalize them so that all rows sum to one.

We combine the **UT**, **IT** and **R** matrix in the transition matrix **A**, as shown in Figure 1. In this model $\alpha \in [0, 1]$ is the weight of the self transitions. Because the sub-matrices are normalized, the rows of **A** also sum to 1, so that they can be used as transition probabilities.

In the initial state vector, two starting points are assigned: $\mathbf{v}_0(u_k) = 1 - \theta$ and $\mathbf{v}_0(t_m) = \theta$ (where u_k is the target user and t_m indicates the selected tag). The parameter θ ($\theta \in [0, 1]$) determines the influence of the personal profile versus the query tag. The state probabilities after n steps are computed by repeating the multiplication of the state vector and the transition matrix **A**. After n steps, the content ranking is obtained by ordering the part of $\mathbf{v}_n$ that corresponds to content ($\mathbf{v}_n(K+1, \ldots, K+L)$) according to the state probabilities. This ranking will also contain the training data (i.e. the items already rated by the target user). We assume that a different user interface is used to browse previously seen content (the user's library), therefore we remove the training examples from the final ranking.

2.1 Self transition (α) and Walk length (n)

Depending on the number of steps in the random walk (n) the final ranking is mostly influenced by the starting points (target user and query tag) or the background distribution. The influence of the background after a certain number of steps is determined by the self-transition probability α. A large self transition probability allows the walk to stay in place (by taking many *self steps*), reinforcing the importance of the starting point, where a small value of α results in a walk that quickly converges to the stable background distribution.

Figure 2 shows the fraction of non-self steps for $\alpha = 0.2$ and $\alpha = 0.8$ at $n = 13$. Because all nodes have the same self transition probability, the total number of non-self steps (Q) after n steps through the social graph is a binomial random variable with the probability mass function (PMF):

$$P_Q(q) = \begin{cases} \binom{n}{q}\alpha^q(1-\alpha)^{n-q} & q = 0, \ldots, n \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Where $P_Q(q)$ is the probability of q non-self steps ($Q = q$).

The PMF shows that if a large value is chosen for α, most of the probability mass will stay close to the starting point and a long tail is created toward more distant nodes. With a small self transition probability the walk quickly moves away from the initial state. We choose a relatively high value of $\alpha = 0.8$ in our experiments to create a slow diffusion of the walk, because we expect to find the most relevant content close to the query, and we want enrich the ranking with a slow integration of more distantly related concepts.

2.2 Query weight (θ)

Most tag-based retrieval systems use the selected tag as query term and rank the content according to popularity ($P(i_l|t_m)$) or freshness ($P(i_l|time)$). Experience from the field of information retrieval has shown that a single term is often not semantically expressive enough to clearly define the user's content need. Our model enriches the query tag by integrating the users history in the search. In the initial state vector, both the query tag and the target user are assigned a value according to θ. The weight of this parameter determines the strength of the personalization. When θ is set to 0, the state probabilities only depend on the profile of the target user, so the predicted content ranking will not be relevant to the query, which closely resembles traditional collaborative filtering [14]. When $\theta = 1$ the state probabilities depend only on the selected query tag, so the result will not be personalized for u_k. If $0 < \theta < 1$ the model derives the probabilities, based on both the target user and the query.

3. DATA

3.1 LibraryThing

LibraryThing[6] is an on-line web service that allows users to create a catalog of the books they own or have read. A user can tag and rate all the books he adds to his personal library. The social aspects of this network give the user the opportunity to meet like-minded people and find new books that match his preference. The popularity of the system has resulted in a database that contains almost 3 million unique works, collaboratively added by more than 300,000 users. We are not aware of any other open network with this amount of collaborative tags ($\approx$ 30 million) and ratings ($\approx$ 3.5 million).

We have collected a trace from the LibraryThing network, containing 25,295 actively tagging users[7]. As expected in data organized by human-activity, we see that the number of books in the users' catalogs follows a power-law distribution [1] (see Fig. 3a). After pruning this data set we retain 7279 users that have all supplied both ratings and tags to at least 20 books. We remove books and tags that occur in less than 5 user profiles, resulting in 37,232 unique works and 10,559 unique tags. This pruned data set contains 2,056,487 UIT relations, resulting in a density of $7.2 * 10^{-7}$ (fraction of non empty cells in **D**). The derived **R**, **UT** and **IT** matrices have a density of respectively: $2.8 \cdot 10^{-3}$, $5.2 \cdot 10^{-3}$ and

[6]http://www.librarything.com
[7]Crawled in July 2007

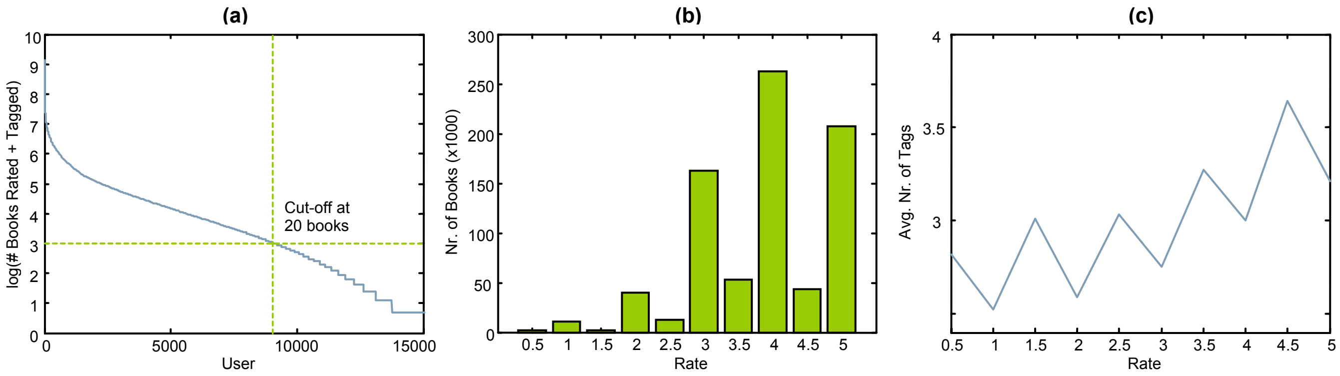

Figure 3: LibraryThing data statistics: a) The number of rated and tagged books stored in the users' catalogs, sorted by size. b) The distribution of rating occurrences in the pruned data set. c) The average number of tags assigned, given the rating.

$2.0 \cdot 10^{-3}$, and all show the power-law behavior common to social networks [6]. We expect that this data is comparable to collaboratively annotated movies, as books and movies comprise the same themes and storylines that can be categorized by tags.

The user interface of LibraryThing allows users to assign ratings on the scale from a half to five. Half ratings can be given by clicking a star twice. The distribution in Figure 3b shows that half ratings occur about 4 times less frequently than whole ratings. Figure 3c shows the relation between the rating and the number of tags given to an item. The upward trend shows that there is a slight correlation between these two variables. This graph also shows that books with half ratings tend to get more tags. This might indicate that the half ratings are used by people who put more effort in the categorization of their books.

4. EXPERIMENTAL SETUP

4.1 Data preparation

In order to estimate the performance of our model without overfitting to the data, we split the data in two equal parts (see Figure 4). Together with all the created annotations (ratings and tags), half of the users (3640 profiles) are put into the *training* set and the other half constitute the *test* set (*Step 1*). We now use the training set to optimize the model parameters by holding out 1/5 of the items of 1/5 of the training users (the validation set). We use our model to predict the held-out content (*Step 2*) using the tags assigned by the target user as query for the content he applied the tag on. Here we assume that the tags assigned to an item by the target user are the same words he would use as query to find the content. For each tag used by u_k we compute the NDCG measure discussed in the next section and compute the mean score over all validation users in the training set (*Step 3*, Figure 5 and 6).

The optimal model parameters derived from the training set are used to compute the performance on the test set, by holding again 1/5 of the user profiles of 1/5 of the users out, and computing the NDCG (*Step 4*). Finally we compare the results of our optimal model to the results achieved with conventional methods (*Step 5*, Table 1 and 2).

4.2 NDCG evaluation

To evaluate the predicted content ranking, we use the Normalized Discounted Cumulative Gain (NDCG) proposed by Järvelin and Kekäläinen [8].

In the predicted content ranking, the rank positions of the held-out validation ratings that correspond to a positive opinion $r \in \{3, 3.5, 4, 4.5, 5\}$ are assigned a value of respectively $G \in \{1, 2, 3, 4, 5\}$, called the *gain*. We do not normalize the rating profiles before assigning the gain, because we expect that the high offset in the ratings (See Figure 3b) is due to the fact that people tend to carefully select the books they read. As a result, people have read many more books they like than books they do not like.

In order to progressively reduce the gain of lower ranked test items, each position in the gain vector is discounted by the $^2\log$ of its index (where we first add 1 to the index, to ensure discounting for all rank positions > 0). The Discounted Cumulative Gain (DCG) now accumulates the values of the discounted gain vector:

$$\text{DCG}[i] = \text{DCG}[i-1] + G[i]/{}^2\log(i+1) \tag{4}$$

The DCG vector is normalized to the optimal DCG vector. This optimal DCG is computed using a gain vector where all test rates are placed in the top of the ranking in descending order. Component by component division now gives us the NDCG vector in which each position contains a value in the range $[0, 1]$ indicating the level of perfectness of the ranking so far. We use the area below the NDCG curve as score to evaluate our rank prediction.

5. EXPERIMENTS

We will discuss the performance of our model on both collaborative and individual tagging systems. In all experiments we fix the self-transition probability (α) to 0.8.

5.1 Collaborative tagging

Parameter optimization To find the optimal model parameters and evaluate the sensitivity of the model we use our random walk to predict the left-out content of the training part of the LibraryThing data. Figure 5 shows the effect of the personalization at different walk lengths. The optimal NDCG is found at $\theta = 0.6$, which means that personalized retrieval gives a more accurate prediction than both completely personal and completely tag based queries.

We also find that the optimal number of steps is larger than one ($n_{optimal} = 13$), which means that the random walk improves a content ranking based on direct relations. Content that has not been tagged extensively will often miss

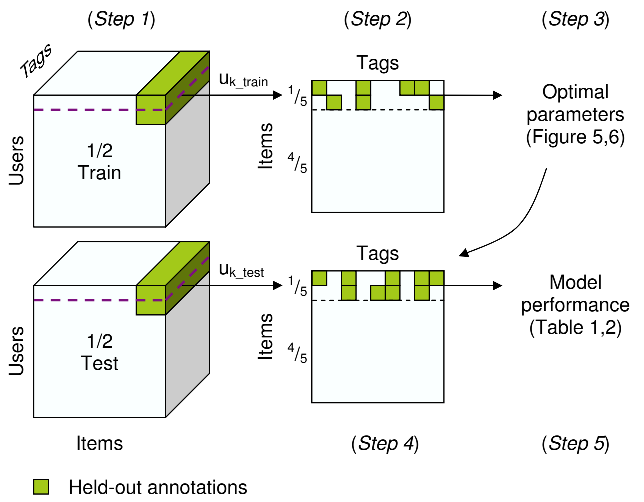

Figure 4: *Step 1*: Splitting of the $\mathbf{D}$ matrix, the $\mathbf{R}$ matrix is split accordingly. *Step 2,4*: A slice of the matrix contains a single user's items and tags. The tags used by that user are in turn used to predict the held-out content.

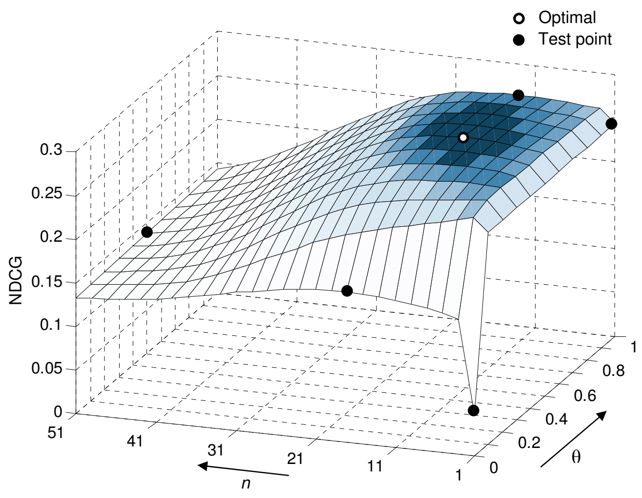

Figure 5: Optimization of the personalization influence θ and the walk length n. The optimal parameters and other test points are indicated with small circles. Note that the NDCG at $n = 1$ and $\theta = 0$ is equal to a random ranking, because a user has no direct link to potentially interesting content.

the terms used as a query by other people. The random walk can find these latent relations that are not explicitly present in the data.

The performance of the model is not very sensitive to small variations in both parameters. Because of the large self transition probability (α), the prediction slowly responds to variations in walk length. Only if the influence of the query tag is completely removed ($\theta \to 0$), the NDCG quickly drops to the global popularity score.

Test results To evaluate our model performance without overfitting to the data, we use a separate test as discussed in Section 4.1. We define two baseline methods: *Random*: The NDCG for a random ranking, and *Global Popularity*: The NDCG at $n = 51$ (We assume that the state vector is fully converged, so that $\mathbf{v}_{51} \approx \mathbf{v}_{\infty}$). We now compare four different model settings, derived from Figure 5. *Popularity Search*: Taking one step in our random walk model ($n = 1$) with $\theta = 1$ gives the ranking according to the number of times the tag was applied to the data. This parameter setting represents the tag browsing as implemented in many social content systems. *Random Walk Search*: Using the optimal number of steps at $\theta = 1$ represents the optimal performance with our model without personalization. Compared to popularity search, this method integrates more indirectly related concepts. *Recommendation*: When the model is completely personal ($\theta = 0$) the ranking will not depend on the query. Obviously this model setting gives lower performance, we see however that the performance is higher than the popularity ranking, indicating a strong coherence within the users' libraries. *Personalized*: The optimal parameter setting of our model.

Table 1: Collaborative tagging: Results on the test set

Model	θ	n	NDCG
Random	-	-	0.0466
Global Popularity	-	51	0.1574
Popularity Search	1	1	0.2378
RW Search	1	13	0.2591
Recommendation	0	17	0.1639
Personalized	0.6	13	0.2642
No Rating	0.6	13	0.2634

The results show that both personalization and smoothing with indirectly related concepts by the random walk improve over traditional tag-based retrieval (See Table 1). Our *personalized* search model outperforms the usual *popularity search* by 11.1% and the random walk model without personalization (*Random walk search*) gives a gain of 9.0%.

Because much related work has used the social graph based on tagging information only [7, 10, 12], we have also optimized our model on the graph created with the $\mathbf{UI}$ matrix instead of the $\mathbf{R}$ matrix. We find that the performance is not significantly lower than the results with our model (Table 1, *No Rating*). This was already indicated by the correlation we found between the rating and the number of assigned tags (Figure 3c). We do however expect that in a data set with more negative opinions, the integration of the explicit preference information might give more performance gain over tag-based user-item relations, because it is impossible to assign a negative amount of tags.

5.2 Individual tagging

To evaluate the benefit of a collaboratively annotated collection over individual tagging we adapt our data by removing all collaborative tags. For each book we randomly select one of its readers and keep only his tags to construct the graph. We use the tags that would be assigned by the other readers as their queries to retrieve the held-out content.

Parameter optimization The results on the training set are shown in Figure 6. We see that the optimal result is shifted to a higher value of θ, meaning that the required influence of personalization is much smaller. Also, a longer walk is needed to reach the optimal value. This can be explained by the fact that the reduced number of edges makes it harder to reach a large amount of relevant content in a small number of steps.

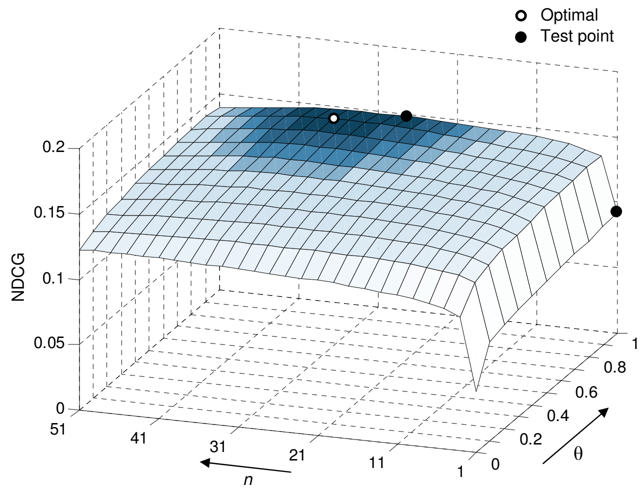

Figure 6: Optimization of the personalization influence θ in an individual tagging system.

Test results We show the results on the test set in Table 2. The NDCG gain of the personalized model over the non-personal model (*Random walk search*) is much smaller compared to the previously discussed results on collaboratively tagged data. In an individual tagging system the user profiles are very limited, because most users will have used only few tags or no tags at all. Therefore, the users' preference is mainly expressed by their given ratings, which appears to be a less informative representation with respect to focused retrieval.

Compared to results on collaborative tagging, the random walk on the individually tagged graph has more improvement over *popularity search* (61.9% improvement). This can be explained, as direct relations are extremely sparse and the random walk smoothly integrates more distantly related concepts. Popularity search performs even worse than a recommendation based on global popularity. Because users associate different terms with specific content, the retrieval model should take latent semantic relations into account, especially in individual tagging systems.

If we remove the rating information and create the social graph with the tag-based **UI** matrix, we see a significant performance drop (Table 2, *No Rating*). In individual tagging systems, the rating information is much more important because it allows people to create direct links with all content, instead of just the injected content.

Table 2: Individual tagging: Results on the test set

Model	θ	n	NDCG
Popularity Search	1	1	0.0926
RW Search	1	27	0.1475
Personalized	0.9	35	0.1499
No Rating	0.9	9	0.1036

6. DISCUSSION

6.1 Related work

A large part of the research on tagging systems has focused on the analysis of statistical patterns arising by the collaborative effort of network users. Golder and Huberman analyzed the structure of social bookmarking in Del.icio.us. They discovered recurring patterns of growth dynamics and identified various user tasks that result in different tagging behavior [5]. Halpin et al. extended this work by investigating the evolution of collaborative tagging patterns into stable distributions by computing the Kulback-Leibler divergence between different time points in Del.icio.us [6]. Marlow et al. showed that individual tagging systems evolve differently over time using data from the popular photo catalog Flickr [11]. Our results demonstrate that individual tagging also drastically reduces retrieval performance, which concurs with the *vocabulary problem* defined by Furnas et al., which states that people tend to use different terms to describe content [4].

Mika extended the bipartite ontology model used in traditional IR by directly integrating the network user in the graph [12]. The resulting tripartite graph gives more insight in the dynamics of social networks. Lambiotte and Ausloos used the same graph to visualize the network structure of Audioscrobbler[8] and CiteULike based on the projected matrices (**UT**, **IT** and **UI**) [10]. Hotho et al. also used the combination of the three binary graphs to apply a variation of adapted PageRank [13] on Del.icio.us data [7]. They only performed an empirical evaluation of their model, making it hard to compare. All these methods use the tag-based **UI** matrix, which does not precisely define the user-item relation. We showed that there is a slight correlation between preference and the number of tags assigned in LibraryThing (Figure 3c). Also, the performance between both information sources does not deviate significantly. However, in individual tagging systems, where most users are not able to apply tags to the content, the ratings provide essential information that can drastically improve content retrieval. We believe that when explicit user preference data is available, this information should be integrated in the social graph, especially in data with few tags or many negative ratings.

Our model is strongly related to the work of Craswell and Szummer [3]. They used a random walk on a query-image graph to retrieve more relevant images for each textual query. Instead of looking at the fully converged state vector ($\mathbf{v}_\infty$) that is used in PageRank related models, they also use the walk length as a model parameter. We have extended their model using the tripartite graph in which users, items and tags constitute the nodes. Because we directly integrate the network user in the model, the tasks that we describe are more focused on social interactions, which meets the desires of many current Internet users. Furthermore, in our model we always start the random walk from the target user, which makes the retrieval task personalized to each user's individual preferences.

6.2 Synonym and homograph robustness

Well known problems in tagging systems are synonyms and homographs. Synonyms are different words that share the same or closely related meaning. The problem in tagging systems arises, because there is no clear regulation on which words to use. If a piece of content has been tagged with a certain word and someone with a different background uses its synonym as a query, the content might not be found. The same problems arise when people use abbreviations, singular or plural words, word combinations and different languages. If a tag cloud is used to query a database, only a

[8]http://www.audioscrobbler.net/

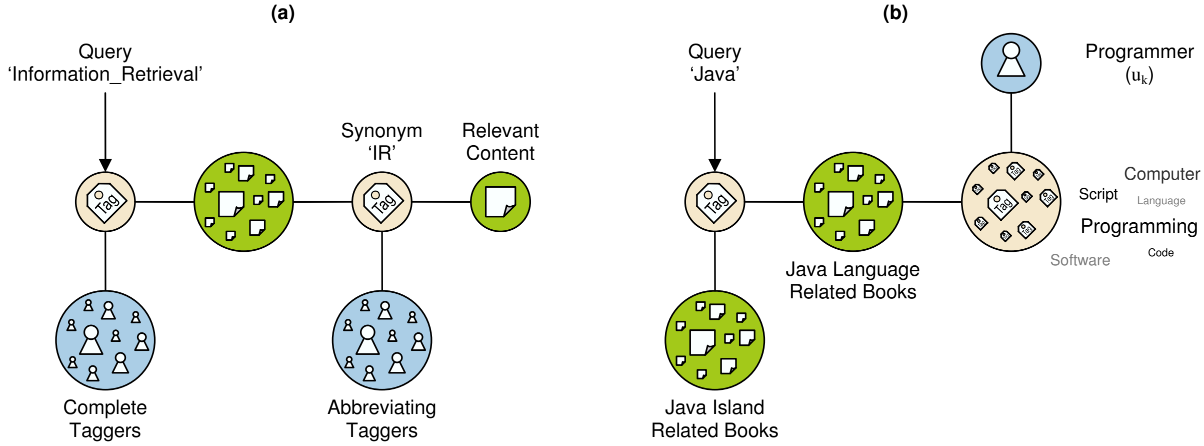

Figure 7: Our model is robust against both synonym and homograph problems: a) Synonyms or spelling differences (like 'Information_Retrieval' and the abbreviation 'IR') reinforce the content ranking because of the soft clustering created by the random walk model. b) Homographs like *Java* can be disambiguated using the target user's history.

single word is used as initial query resulting in sub-optimal retrieval performance.

Clustering methods have been proposed to group tags with strong lexical relations [2]. Clustering algorithms create binary relations between concepts although the natural similarity between words is a continuous relation. A random walk has shown to have a soft clustering effect that smoothly relates similar concepts before converging to the background probability [16]. Figure 7a shows that if enough users have tagged certain content, this soft clustering makes our model robust against synonymity problems, as synonym terms will be connected through a high volume of item connections.

Homographs are words that do not necessarily have the same pronunciation, but are written in exactly the same way. If a browsing user selects a homograph as query, the system will not know which denotation the user aimed for. In order to disambiguate the terms a user is looking for, our model integrates the information about the past behavior of that user. Because our random walk starts at both the query tag and the target user, the content that matches the target user's preference is more likely to be found first (see Figure 7b.).

7. REFERENCES

[1] A.-L. Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211.

[2] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *WWW2006: Proceedings of the Collaborative Web Tagging Workshop*, Edinburgh, Scotland, 2006.

[3] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 239–246, New York, NY, USA, 2007. ACM Press.

[4] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971, November 1987.

[5] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32(2):198–208, April 2006.

[6] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 211–220, New York, NY, USA, 2007. ACM Press.

[7] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *ESWC 2006, LNCS 4011*, pages 411–426. Springer-Verlag Berlin Heidelberg, 2006.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[9] O. Kaser and D. Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *Tagging and Metadata for Social Information Organization Workshop (WWW 2007)*, May 2007.

[10] R. Lambiotte and M. Ausloos. Collaborative tagging as a tripartite network. In *ICCS 2006, LNCS 3993*, pages 1114–1117. Springer-Verlag Berlin Heidelberg, May 2006.

[11] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM Press.

[12] P. Mika. Ontologies are us: A unified model of social networks and semantics. In Y. Gil, E. Motta, R. V. Benjamins, and M. Musen, editors, *ISWC 2005, LNCS 3729*, pages 522–536. Springer-Verlag Berlin Heidelberg, 2005.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[14] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, March 1997.

[15] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[16] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, volume 14, 2001.