A Tag Recommender System Exploiting User and Community Behavior

Cataldo Musto
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
cataldomusto@di.uniba.it

Fedelucio Narducci
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
narducci@di.uniba.it

Marco De Gemmis
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
degemmis@di.uniba.it

Pasquale Lops
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
lops@di.uniba.it

Giovanni Semeraro
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
semeraro@di.uniba.it

ABSTRACT

Nowadays Web sites tend to be more and more social: users can upload any kind of information on collaborative platforms and can express their opinions about the content they enjoyed through textual feedbacks or reviews. These platforms allow users to annotate resources they like through freely chosen keywords (called tags). The main advantage of these tools is that they perfectly fit user needs, since the use of tags allows organizing the information in a way that closely follows the user mental model, making retrieval of information easier. However, the heterogeneity characterizing the communities causes some problems in the activity of social tagging: someone annotates resources with very specific tags, other people with generic ones, and so on. These drawbacks reduce the exploitation of collaborative tagging systems for retrieval and filtering tasks. Therefore, systems that assist the user in the task of tagging are required. The goal of these systems, called tag recommenders, is to suggest a set of relevant keywords for the resources to be annotated. This paper presents a tag recommender system called STaR (Social Tag Recommender system). Our system is based on two assumptions: 1) the more two or more resources are similar, the more they share common tags 2) a tag recommender should be able to exploit tags the user already used in order to extract useful keywords to label new resources. We also present an experimental evaluation carried out using a large dataset gathered from Bibsonomy.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing: Indexing methods; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval: Information filtering

General Terms

Algorithms, Experimentation

Keywords

Recommender Systems, Web 2.0, Collaborative Tagging Systems, Folksonomies

1. INTRODUCTION

We are assisting to a transformation of the Web towards a more user-centric vision called Web 2.0. By using Web 2.0 applications users are able to publish auto-produced contents such as photos, videos, political opinions, reviews, hence they are identified as *Web prosumers*: *pro*ducers + con*sumers* of knowledge. Recently the research community has thoroughly analyzed the dynamics of *tagging*, which is the act of annotating resources with free labels, called *tags*. Many argue that, thanks to the expressive power of *folksonomies* [17], collaborative tagging systems are very helpful to users in organizing, browsing and searching resources. This happens because, in contrast to systems where information about resources is only provided by a small set of experts, the model of collaborative tagging systems takes into account the way individuals conceive the information contained in a resource [18], so they perfectly fit user needs and user mental model. Nowadays almost all Web 2.0 platforms embed tagging: we can cite Flickr[1], YouTube[2], Del.icio.us[3], Last.fm[4], Bibsonomy[5] and so on. These systems provide heterogeneous contents (photos, videos, musical habits, etc.), but they all share a common core: they let users to post new resources and to annotate them with tags. Besides the simple act of annotation, the tagging of resources has also a key social aspect; the connection between users, resources and tags generates a tripartite graph that can be easily exploited to

[1]http://www.flickr.com
[2]http://www.youtube.com
[3]http://delicious.com/
[4]http://www.last.fm/
[5]http://www.bibsonomy.org/

analyze the dynamics of collaborative tagging systems. For example, users that label the same resource by using the same tags might have similar tastes and items labeled with the same tags might share common characteristics.

Undoubtedly the power of tagging lies in the ability for people to freely determine the appropriate tags for a resource [10]. Since folksonomies do not rely on a predefined lexicon or hierarchy they have the main advantage to be fully free, but at the same time they generate a very noisy tag space, really hardly to exploit for retrieval or recommendation tasks without performing any form of processing. Golder et. al. [4] identified three major problems of collaborative tagging systems: *polysemy*, *synonymy*, and *level variation*. Polysemy refers to situations where tags can have multiple meanings: for example a resource tagged with the term *bat* could indicate a news taken from an online sport newspaper or a Wikipedia article about nature. We refer to synonymy when multiple tags share a single meaning: for example we can have simple morphological variations (such as 'AI','artificial_intelligence' and so on to identify a scientific publication about Artificial Intelligence) but also lexical relations (like resources tagged with 'arts' versus 'cultural heritage'). At last, the phenomenon of tagging at different levels of abstraction is defined as *level-variation*. This happens when people annotate the same web page, containing for example a recipe for roast turkey with the tag 'roast-turkey' but also with a simple 'recipe'.

Since these problems are of hindrance to completely exploit the expressive power of folksonomies, in the last years many tools have been developed to assist the user in the task of tagging and to aid at the same time the tag convergence [3]: we refer to them as tag recommenders. These systems work in a very simple way:

1. a user posts a resource;
2. depending on the approach, the tag recommender analyzes some information related to the resource (usually metadata or a subset of the relations in the aforementioned tripartite graph);
3. the tag recommender processes this information and produces a list of recommended tags;
4. the user freely chooses the most appropriate tags to annotate the resource.

Clearly, the more these recommended tags match the user needs and her mental model, the more she will use them to annotate the resource. In this way we can rapidly speed up the tag convergence aiding at the same time in filtering the noise of the complete tag space.
This paper presents the tag recommender STaR. When developing the model, we tried to point out two concepts:

- resources with similar content should be annotated with similar tags;
- a tag recommender needs to take into account the previous tagging activity of users, increasing the weight of the tags already used to annotate similar resources.

In this work, we identify two main aspects in the tag recommendation task: first, each user has a typical manner to label resources; second, similar resources usually share common tags.

The paper is organized as follows. Section 2 analyzes related work. Section 3 explains the architecture of the system and how the recommendation approach is implemented. The experimental evaluation carried out is described in Section 4, while conclusions and future work are drawn in the last section.

2. RELATED WORK

Usually the works in the tag recommendation area are broadly divided into three classes: *content-based*, *collaborative* and *graph-based* approaches.

In the content-based approach, exploiting some Information Retrieval-related techniques, a system is able to extract relevant unigrams or bigrams from the text. Brooks et. al [2], for example, develop a tag recommender system that exploits TF/IDF scoring [13] in order to automatically suggests tags for a blog post. In [5] is presented a novel method for key term extraction from text documents. Firstly, every document is modeled as a graph which nodes are terms and edges represent semantic relationship between them. These graphs are then partitioned using communities detection techniques and weighted exploiting information extracted from Wikipedia. The tags composing the most relevant communities (a set of terms related with the topic of the resource) are then suggested to the user.

AutoTag [11] is one of the most important systems implementing the collaborative approach for tag recommendation. It presents some analogies with collaborative filtering methods: as in the collaborative recommender systems the recommendations are generated based on the ratings provided by similar users (called neighbors), in AutoTag the system suggests tags based on the other tags associated with similar posts. Firstly, the tool exploits some IR techniques in order to find similar posts and extracts the tags they are annotated with. All the tags are then merged, building a folksonomy that is filtered and re-ranked. The top-ranked tags are finally suggested to the user, who selects the most appropriate ones to attach to the post.

TagAssist [15] extends the AutoTags' approach introducing some preprocessing step (specifically, a lossless compression over existing data) in order to improve the quality of the recommendations. The core of this approach is represented by a a Tag Suggestion Engine (TSE) which leverages previously tagged posts providing appropriate suggestions for new content.

Marinho [9] investigates the user-based collaborative approach for tag recommendation. The main outcome of this work shows that users with a similar tag vocabulary tend to tag alike, since the method seems to produce good results when applied on the user-tag matrix.

The problem of tag recommendation through graph-based approaches has been firstly addressed by Jäschke et al. in [7]. The key idea behind their FolkRank algorithm is that a resource which is tagged by important tags from important users becomes important itself. So, they build a graph whose nodes mutually reinforces themselves by spreading their weights. They compared some recommendation techniques including collaborative filtering, PageRank and FolkRank, showing that the FolkRank algorithm outperforms other approaches. Furthermore, Schmitz et al. [14] proposed association rule mining as a technique that might be useful in the tag recommendation process.

In literature we can find also other methods (called *hy-*

brid), which try to integrate two of more sources of knowledge (mainly, content and collaborative ones) in order to improve the quality of recommended tags.

Heymann et. al [6] present a tag recommender that exploits at the same time social knowledge and textual sources. They produce recommendations exploiting both the HTML source code (extracting anchor texts and page texts) and the annotations of the community. The effectiveness of this approach is also confirmed by the use of a large dataset crawled from del.icio.us for the experimental evaluation.

Lipczak in [8] proposes a similar hybrid approach. Firstly, the system extracts tags from the title of the resource. Afterwards, it performs an analysis of co-occurrences, in order to expand the sets of candidate tags with the tags that usually co-occur with terms in the title. Finally, tags are filtered and re-ranked exploiting the informations stored in a so-called "personomy", the set of the tags previously used by the user.

Finally, in [16] the authors proposed a model based on both textual contents and tags associated with the resource. They introduce the concept of *conflated tags* to indicate a set of related tags (like blog, blogs, ecc.) used to annotate a resource. Modeling in this way the existing tag space they are able to suggest various tags for a given bookmark exploiting both *user* and *document* models.

3. STAR: A SOCIAL TAG RECOMMENDER SYSTEM

Following the definition introduced in [7], a folksonomy can be described as a triple (U, R, T) where:

- U is a set of *users*;
- R is a set of *resources*;
- T is a set of *tags*.

We can also define a tag assignment function TAS: $U \times R \rightarrow T$.

So, a collaborative tagging system is a platform composed of users, resources and tags that allows users to freely assign tags to resources, while the tag recommendation task for a given user $u \in U$ and a resource $r \in R$ can be described as the generation of a set of tags $\text{TAS}(u, r) \subseteq T$ according to some relevance model. In our approach these tags are generated from a ranked set of *candidate tags* from which the top n elements are suggested to the user.

STaR (Social Tag Recommender) is a content-based tag recommender system, developed at the University of Bari. The inceptive idea behind STaR is to improve the model implemented in systems like TagAssist [15] or AutoTag [11].

Although we agree that similar resources usually share similar tags, in our opinion Mishne's approach presents two important drawbacks:

1. the tag re-ranking formula simply performs a sum of the occurrences of each tag among all the folksonomies, without considering the similarity with the resource to be tagged. In this way tags often used to annotate resources with a low similarity level could be ranked first;
2. the proposed model does not take into account the previous tagging activity performed by users. If two users bookmarked the same resource, they will receive the same suggestions since the folksonomies built from similar resources are the same.

We will try to overcome these drawbacks, by proposing an approach firstly based on the analysis of similar resources capable also of leveraging the tags already selected by the user during her previous tagging activity, by putting them on the top of the tag rank. Figure 1 shows the general architecture of STaR. The recommendation process is performed in four steps, each of which is handled by a separate component.

3.1 Indexing of Resources

Given a collection of resources (*corpus*) with some textual metadata (such as the title of the resource, the authors, the description, etc.), STaR firstly invokes the *Indexer* module in order to perform a preprocessing step on these data by exploiting Apache Lucene[6]. Obviously, the kind of metadata to be indexed is strictly dependant on the nature of the resources. For example, supposing to recommend tags for *bookmarks*, we could index the title of the web page and the extended description provided by users, while for *BibteX* entries, we could index the title of the publication and the abstract. Let U be the set of users and N the cardinality of this set, the indexing procedure is repeated $N + 1$ times: we build an index for each user (*Personal Index*) storing the information on the resources she previously tagged and an index for the whole community (*Social Index*) storing the information about all the tagged resources by merging the singles Personal Indexes.

Following the definitions presented above, given a user $u \in U$ we define $PersonalIndex(u)$ as:

$$PersonalIndex(u) = \{r \in R | \exists t \in T : \text{TAS}(u, r) = t\} \quad (1)$$

where TAS is the tag assignment function TAS: $U \times R \rightarrow T$ which assigns tags to a resource annotated by a given user. *SocialIndex* represents the union of all the user personal indexes:

$$SocialIndex = \bigcup_{i=1}^{N} PersonalIndex(u_i) \quad (2)$$

3.2 Retrieval of Similar Resources

Next, STaR can take into account users requests in order to produce personalized tag recommendations for each resource. First, every user has to provide some information about the resource to be tagged, such as the title of the Web page or its URL, in order to crawl the textual metadata associated on it.

Next, if the system can identify the user since she has already posted other resources, it exploits data about her (language, the tags she uses more, the number of tags she usually uses to annotate resources, etc.) in order to refine the query to be submitted against both the *Social* and *Personal* indexes stored in Lucene. We used as query the title of the web page (for bookmarks) or the title of the publication (for BibTeX entries). Obviously before submitting the query we processed it by deleting not useful characters and punctuation.

In order to improve the performances of the Lucene Querying Engine we replaced the original Lucene Scoring function

[6]http://lucene.apache.org

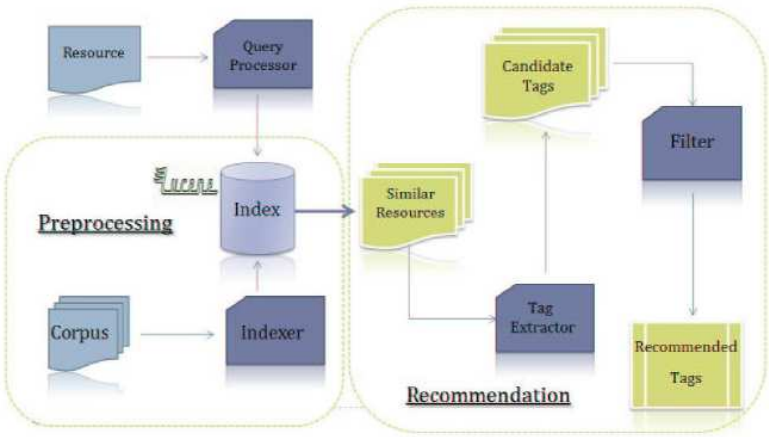

Figure 1: Architecture of STaR

with an Okapi BM25 implementation[7]. BM25 is nowadays considered as one of the state-of-the art retrieval models by the IR community [12].

Let D be a corpus of documents, $d \in D$, BM25 returns the top-k resources with the highest similarity value given a resource r (tokenized as a set of terms $t_1 \dots t_m$), and is defined as follows:

$$sim(r,d) = \sum_{i=1}^{m} \frac{n^r_{t_i}}{k_1((1-b)+b*l)+n^r_{t_i}} * idf(t_i) \quad (3)$$

where $n^r_{t_i}$ represents the occurrences of the term t_i in the document d, l is the ratio between the length of the resource and the average length of resources in the corpus. Finally, k_1 and b are two parameters typically set to 2.0 and 0.75 respectively, and $idf(t_i)$ represents the inverse document frequency of the term t_i defined as follows:

$$idf(t_i) = log\frac{N + df(t_i) + 0.5}{df(t_i) + 0.5} \quad (4)$$

where N is the number of resources in the collection and $df(t_i)$ is the number of resources in which the term t_i occurs.

Given user $u \in U$ and a resource r, Lucene returns the resources whose similarity with r is greater or equal than a threshold β. To perform this task Lucene uses both the *PersonalIndex* of the user u and the *SocialIndex*. More formally:

$$P\_Res(u,q) = \{r \in PersonalIndex(u) | sim(q,r) \geq \beta\}$$

$$S\_Res(q) = \{r \in SocialIndex | sim(q,r) \geq \beta\}$$

Figure 2 depicts an example of the retrieving step. In this case the target resource is represented by Gazzetta.it, one of the most famous Italian sport newspaper. Lucene queries the *SocialIndex* and returns as the most similar resources an online newspaper (Corrieredellosport.it) and the official web site of an Italian Football Club (Inter.it). The

[7]http://nlp.uned.es/ jperezi/Lucene-BM25/

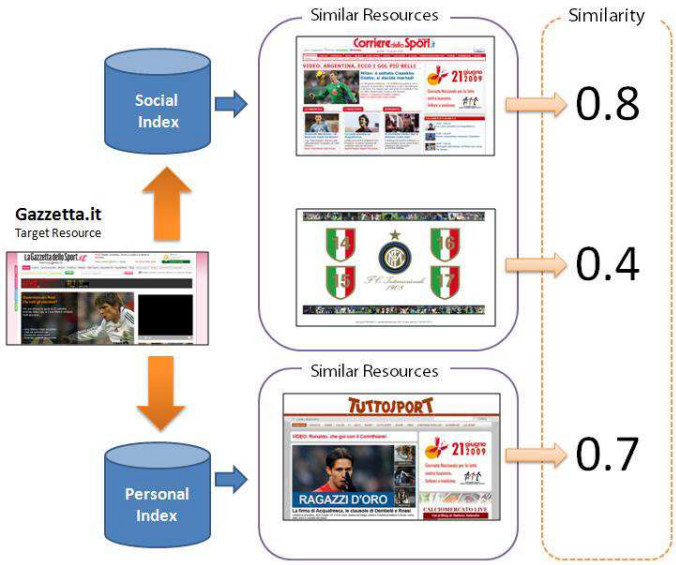

Figure 2: Retrieval of Similar Resources

PersonalIndex, instead, returns another online newspaper (Tuttosport.com). The similarity score returned by Lucene has been normalized.

3.3 Extraction of Candidate Tags

The role of the *Tag Extractor* is to produce as output the list of the so-called "candidate tags" (namely, the tags considered as 'relevant' by the tag recommender). In this step the system gets the most similar resources returned by the Apache Lucene engine and builds their folksonomies (namely, the tags they have been annotated with). Next, it produces the list of candidate tags by computing for each tag from the folksonomy a score obtained by weighting the similarity score returned by Lucene with the normalized occurrence of the tag. If the *Tag Extractor* also gets the list of the most similar resources from the user *PersonalIndex*, it will produce two partial folksonomies that are merged, assigning a weight to each folksonomy in order to boost the

tags previously used by the user.

Formally, for each query q (namely, the resource to be tagged), we can define a set of tags to recommend by building two sets: $candTags_p$ and $candTags_s$. These sets are defined as follows:

$$candTags_p(u,q) = \{t \in T | t = TAS(u,r) \wedge r \in P\_Res(u,q)\}$$

$$candTags_s(q) = \{t \in T | t = TAS(u,r) \wedge r \in S\_Res(q) \wedge u \in U\}$$

In the same way we can compute the relevance of each tag with respect to the query q as:

$$rel_p(t,u,q) = \frac{\sum_{r \in P\_Res(u,q)} n_r^t * sim(r,q)}{n^t} \quad (5)$$

$$rel_s(t,q) = \frac{\sum_{r \in S\_Res(q)} n_r^t * sim(r,q)}{n^t} \quad (6)$$

where n_r^t is the number of occurrences of the tag t in the annotation for resource r and n^t is the sum of the occurrences of tag t among all similar resources.

Finally, the set of *Candidate Tags* can be defined as:

$$candTags(u,q) = candTags_p(u,q) \cup candTags_s(q) \quad (7)$$

where for each tag t the global relevance can be defined as:

$$rel(t,q) = \alpha * rel_p(t,q) + (1-\alpha) * rel_s(t,q) \quad (8)$$

where α (PersonalTagWeight) and $(1-\alpha)$ (SocialTagWeight) are the weights of the personal and social tags respectively.

Figure 3 depicts the procedure performed by the *Tag Extractor*: in this case we have a set of 4 Social Tags (Newspaper, Online, Football and Inter) and 3 Personal Tags (Sport, Newspaper and Tuttosport). These sets are then merged, building the set of *Candidate Tags*. This set contains 6 tags since the tag *newspaper* appears both in social and personal tags. The system associates a score to each tag that indicates its effectiveness for the target resource. Besides, the scores for the Candidate Tags are weighted again according to SocialTagWeight (α) and PersonalTagWeight ($1-\alpha$) values (in the example, 0.3 and 0.7 respectively), in order to boost the tags already used by the user in the final tag rank. Indeed, we can point out that the social tag 'football' gets the same score of the personal tag 'tuttosport', although its original weight was twice.

3.4 Tag Recommendation

Finally, the last step of the recommendation process is performed by the *Filter*. It removes from the list of candidate tags the ones not matching specific conditions, such as a threshold for the relevance score computed by the Tag Extractor. Obviously, the value for the threshold and the maximum number of tags to be recommend is strictly dependent from the training data.

Formally, given a user $u \in U$, a query q and a threshold value γ, the goal of the filtering component is to build $rec(u,q)$ defined as follows:

$$rec(u,q) = \{t \in candTags(u,q) | rel(t,q) > \gamma\}$$

Table 1: Results comparing the Lucene original scoring function with BM25

Scoring	Resource	Pr	Re	F1
Original	bookmark	25.26	29.67	27.29
BM25	bookmark	25.62	36.62	30.15
Original	bibtex	14.06	21.45	16.99
BM25	bibtex	13.72	22.91	17.16
Original	overall	16.43	23.58	19.37
BM25	overall	**16.45**	**26.46**	**20.29**

In the example in Figure 3, setting a threshold $\gamma = 0.20$, the system would suggest the tags *sport* and *newspaper*.

4. EXPERIMENTAL EVALUATION

We designed two different experimental sessions to evaluate the performance of the tag recommender. In the first session we performed a comparison between the original scoring function of Lucene and a novel BM25 implementation, while the second was carried out to tune the system parameters.

4.1 Description of the dataset

We designed the experimental evaluation by exploiting a dataset gathered from Bibsonomy. It contains *263,004 bookmark* posts and *158,924 BibTeX* entries submitted by 3,617 different users. For each of the 235,328 different URLs and the 143,050 different BibTeX entries were also provided some textual metadata (such as the title of the resource, the description, the abstract and so on).

We evaluated STaR by comparing the real tags (namely, the tags a user adopts to annotate an unseen resource) with the suggested ones. The accuracy was finally computed using classical IR metrics, such as Precision, Recall and F1-Measure. *Precision* (Pr) is defined as the number of relevant recommended tags divided by the number of recommended tags. *Recall* (Re) is defined as the number of relevant recommended tags divided by the total number of relevant tags available. The F1-measure is computed by the following formula:

$$F1 = \frac{(2 * Pr * Re)}{Pr + Re} \quad (9)$$

4.2 Experimental Session 1

Firstly, we tried to evaluate the predictive accuracy of STaR comparing difference scoring function (namely, the Lucene original one and the aforementioned BM25 implementation). We performed the same steps previously described, retrieving the most similar items using the two mentioned similarity functions and comparing the tags suggested by the system in both cases. Results are presented in Table 1.

In general, there is an improvement by adopting BM25 with respect to the Lucene original similarity function. We can note that BM25 improved the both the recall (+ 6,95% for bookmarks, +1,46% for BibTeXs entries) and the F1 measure (+ 2,86% for bookmarks, +0,17% for BibTeXs entries).

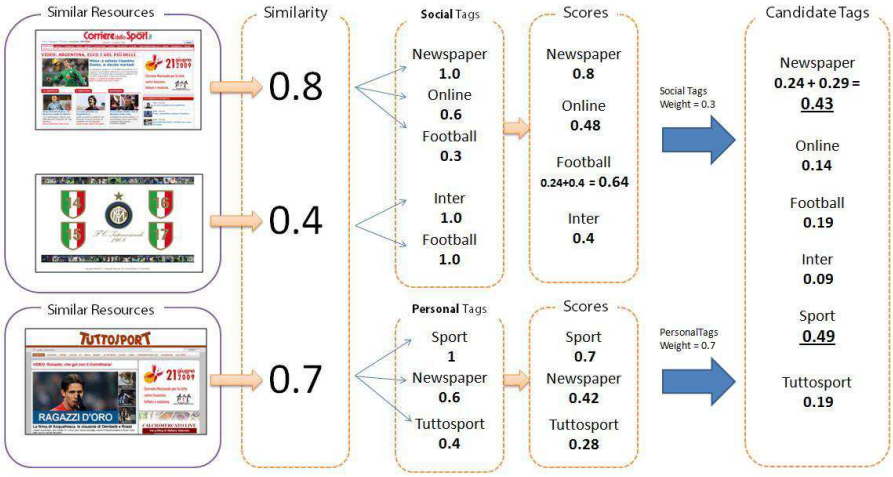

Figure 3: Description of the process performed by the Tag Extractor

4.3 Experimental Session 2

Next we designed a second experimental evaluation in order to compare the predictive accuracy of STaR with different combinations of system parameters. Namely:

- the maximum number of similar documents retrieved by Lucene;
- the value of α for the *PersonalTagWeight* and *SocialTagWeight* parameters;
- the threshold γ to establish whether a tag is relevant;
- which fields of the target resource use to compose the query;
- the best scoring function between Lucene standard one and Okapi BM25.

First, tuning the number of similar documents to retrieve from the *PersonalIndex* and *SocialIndex* is very important, since a value too high can introduce noise in the retrieval process, while a value too low can exclude documents containing relevant tags. By analyzing the results returned by some test queries, we decided to set this value between 5 and 10, depending on the training data.

Next, we tried to estimate the values for *PersonalTagWeight* (PTW) and the *SocialTagWeight* (STW). An higher weight for the Personal Tags means that in the recommendation process the systems will weigh more the tags previously used by the target user, while an higher value for the Social Tags will give more importance to the tags used by the community (namely, the whole folksonomy) on the target resource. These parameters are biased by the user practice: if tags often used by the user are very different from those used from the community, the PTW should be higher than STW. We performed an empirical study since it is difficult to define the user behavior at run time. We tested the system setting the parameters with several combinations of values:

i) PTW = 0.7 STW = 0.3;
ii) PTW = 0.5 STW = 0.5;
iii) PTW = 0.3 STW = 0.7.

Another parameter that can influence the system performance is the set of fields to use to compose the query. For each resource in the dataset there are many textual fields, such as title, abstract, description, extended description, etc. In this case we used as query the title of the webpage (for bookmarks) and the title of the publication (for BibTeX entries).

The last parameter we need to tune is the threshold to deem a tag as relevant (γ).We performed some tests suggesting both 4 and 5 tags and we decided to recommend only 4 tags since the fifth was usually noisy. We also fixed the threshold value between 0.20 and 0.25.

In order to carry out this experimental session we used the aforementioned dataset both as training and test set. We executed the test over 50,000 bookmarks and 50,000 BibTeXs. For each resource randomly chosen from the dataset and for each combination of parameters, we executed the following steps:

- query preparation;
- Lucene retrieval function invocation;
- building of the set of Candidate Tags;
- comparing the recommended tags with the real tags associated by the user;
- computing of Precision, Recall, and F1-measure.

Results are presented in Table 2 and Table 3.

Analyzing the results (see Figure **??**), it emerges that the approach we called *user-based* outperformed the other ones. In this configuration we set PTW to 1.0 and STW to 0, so we suggest only the tags already used by the user in tagging similar resources. No query was submitted against the *SocialIndex*. The first remark we can make is that each user has her own mental model and her own vocabulary: she usually prefers to tag resources with labels she already used. Instead, getting tags from the *SocialIndex* only (as proved by the results of the community-based approach) often introduces some noise in the recommendation process. The hybrid approaches outperformed the community-based one, but their predictive accuracy is still worse when compared with the user-based approach. Finally, all the approaches

Table 2: Predictive accuracy of STaR over 50,000 bookmarks

Approach	STW	PTW	Pr	Re	F1
Comm.-based	1.0	0.0	23.96	24.60	24.28
User-based	0.0	1.0	**32.12**	**28.72**	**30.33**
Hybrid	0.7	0.3	24.96	26.30	25.61
Hybrid	0.5	0.5	24.10	25.16	24.62
Hybrid	0.3	0.7	23.85	25.12	25.08
Baseline	-	-	35.58	10.42	16.11

Table 3: Predictive accuracy of STaR over 50,000 BibTeXs

Approach	STW	PTW	Pr	Re	F1
Comm.-based	1.0	0.0	34.44	35.89	35.15
User-based	0.0	1.0	**44.73**	**40.53**	**42.53**
Hybrid	0.7	0.3	32.31	38.57	35.16
Hybrid	0.5	0.5	32.36	37.55	34.76
Hybrid	0.3	0.7	35.47	39.68	37.46
Baseline	-	-	42.03	13.23	20.13

outperformed the F1-measure of the baseline. We computed the baseline recommending for each resource only its most popular tags. Obviously, for resources never tagged we could not suggest anything.

This analysis substantially confirms the results we obtained from other studies performed in the area of the tag-based recommendation [1].

5. CONCLUSIONS AND FUTURE WORK

Collaborative Tagging Systems are powerful tools, since they let users to organize the information in a way that perfectly fits their mental model. However, typical drawbacks of collaborative tagging systems represent an hindrance, since the complete tag space is too noisy to be exploited for retrieval and filtering task. So, systems that assist users in the task of tagging speeding up the tag convergence are more and more required. In this paper we presented STaR, a social tag recommender system. The idea behind our work was to discover similarity among resources in order to exploit communities and user tagging behavior. In this way our recommender system was able to suggest tags for users and items still not stored in the training set. The experimental sessions showed that users tend to reuse their own tags to annotate similar resources, so this kind of recommendation model could benefit from the use of the user personal tags before extracting the social tags of the community (we called this approach user-based). Next, we showed that the integration of a more effective scoring function (BM25) could also improve the overall accuracy of the system.

This approach has a main drawback, since it cannot suggest any tags when the set of similar items returned by Lucene is empty. So, we plan to extend the system in order to extract significant keywords from the textual content associated to a resource (title, description, etc.) that has not similar items, maybe exploiting structured data or domain ontologies. Furthermore, since tags usually suffer of typical Information Retrieval problem (namely, polysemy, synonymy, etc.) we will try to establish if the integration of Word Sense Disambiguation tools or a semantic representation of documents could improve the performance of recommender. Another issue to analyze is the application of our methodology in different domains such as multimedia environment. In this field discovering similarity among items just on the ground of textual content could be not sufficient. Finally, we will perform also some studies in the area of tag-based recommendation, investigating the integration of tag recommenders for recommendations tasks, since reaching more quickly the tag convergence could help to build better folksonomies and to produce more accurate recommendations.

6. REFERENCES

[1] P. Basile, M. de Gemmis, P. Lops, G. Semeraro, M. Bux, C. Musto, and F. Narducci. FIRSt: a Content-based Recommender System Integrating Tags for Cultural Heritage Personalization. In P. Nesi, K. Ng, and J. Delgado, editors, *Proceedings of the 4th International Conference on Automated Solutions for Cross Media Content and Multi-channel Distribution (AXMEDIS 2008) - Workshop Panels and Industrial Applications, Florence, Italy*, Firenze University Press, pages 103–106, November 17-19, 2008.

[2] C. H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 625–632, New York, NY, USA, 2006. ACM Press.

[3] C. Cattuto, C. Schmitz, A. Baldassarri, V. D. P. Servedio, V. Loreto, A. Hotho, M. Grahl, and G. Stumme. Network properties of folksonomies. *AI Communications*, 20(4):245–262, December 2007.

[4] S. Golder and B. A. Huberman. The Structure of Collaborative Tagging Systems. *Journal of Information Science*, 32(2):198–208, 2006.

[5] Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. Extracting key terms from noisy and multi-theme documents. In *18th International World Wide Web Conference*, pages 651–661, April 2009.

[6] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[7] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In Alexander Hinneburg, editor, *Workshop Proceedings of Lernen - Wissensentdeckung - Adaptivit?t (LWA 2007)*, pages 13–20, September 2007.

[8] M. Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 84–95, 2008.

[9] L. B. Marinho and L. Schmidt-Thieme. Collaborative tag recommendations. pages 533–540. 2008.

[10] A. Mathes. Folksonomies - cooperative classification

and communication through shared metadata. Website, December 2004. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.

[11] G. Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM.

[12] S. E. Robertson, S. Walker, M. H. Beaulieu, A. Gull, and M. Lau. Okapi at trec. In *Text REtrieval Conference*, pages 21–30, 1992.

[13] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[14] C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Őiberna, editors, *Data Science and Classification (Proc. IFCS 2006 Conference)*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Berlin/Heidelberg, July 2006. Springer. Ljubljana.

[15] S. Sood, S. Owsley, K. Hammond, and L. Birnbaum. TagAssist: Automatic Tag Suggestion for Blog Posts. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[16] M. Tatu, M. Srikanth, and T. D'Silva. Rsdc'08: Tag recommendations using bookmark content. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 96–107, 2008.

[17] T. Vander Wal. Folksonomy coinage and definition. Website, Februar 2007. http://vanderwal.net/folksonomy.html.

[18] H. Wu, M. Zubair, and K. Maly. Harvesting social knowledge from folksonomies. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 111–114, New York, NY, USA, 2006. ACM Press.