An IR-based approach to Tag Recommendation

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

Thanks to the continuous growth of collaborative platforms like YouTube, Flickr and Delicious, we are recently witnessing to a rapid evolution of web dynamics towards a more 'social' vision, called Web 2.0. In this context collaborative tagging systems are rapidly emerging as one of the most promising tools. However, as tags are handled in a simply syntactical way, collaborative tagging systems suffer of typical Information Retrieval (IR) problems like polysemy and synonymy: so, in order to reduce the impact of these drawbacks and to aid at the same time the so-called tag convergence, systems that assist the user in the task of tagging are required.

In this paper we present a system, called STaR, that implements an IR-based approach for tag recommendation. Our approach, mainly based on the exploitation of a state-of-the-art IR-model called BM25, relies on two assumptions: firstly, if two or more resources share some common patterns (e.g. the same features in the textual description), we can exploit this information supposing that they could be annotated with similar tags. Furthermore, since each user has a typical manner to label resources, a tag recommender might exploit this information to weigh more the tags she already used to annotate similar 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

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

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*. 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 analyze the dynamics of collaborative tagging systems. 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 hard to exploit for retrieval or recommendation tasks without performing any form of processing.

This problem is a hindrance to completely exploit the expressive power of folksonomies, so 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: we refer to them as tag recommenders.

This paper presents STaR, a tag recommender system implementing an IR-based approach that relies on a state-of-the-art IR model called BM25. In this work, already presented [5],within the ECML-PKDD 2009 Discovery Challenge[1], 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.

[1]http://www.kde.cs.uni-kassel.de/ws/dc09

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 in order to automatically suggests tags for a blog post.

AutoTag [4] 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.

The problem of tag recommendation through graph-based approaches has been firstly addressed by Jäschke et al. in [3]. The key idea behind their FolkRank algorithm is that a resource which is tagged by important tags from important users becomes important itself. Furthermore, Schmitz et al. [7] proposed association rule mining as a technique that might be useful in the tag recommendation process.

3. STAR: A SOCIAL TAG RECOMMENDER SYSTEM

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 [8] or AutoTag [4].

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.

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[2]. Obviously, the kind of metadata to be indexed is strictly dependent on the nature of the resources. 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 Personal Indexes.

3.2 Retrieval of Similar Resources

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.

In order to improve the performances of the Lucene Querying Engine we replaced the original Lucene Scoring function with an Okapi BM25 implementation[3]. BM25 is nowadays considered as one of the state-of-the art retrieval models by the IR community [6].

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:

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

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:

$$idf(t_i) = log\frac{N - df(t_i) + 0.5}{df(t_i) + 0.5} \qquad (2)$$

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 a user 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*.

For example, we suppose that the target resource is represented by Gazzetta.it, one of the most famous Italian sport newspaper. Lucene queries the *SocialIndex* and it could returns as the most similar resources an online newspaper (Corrieredellosport.it) and the official web site of an Italian

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

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

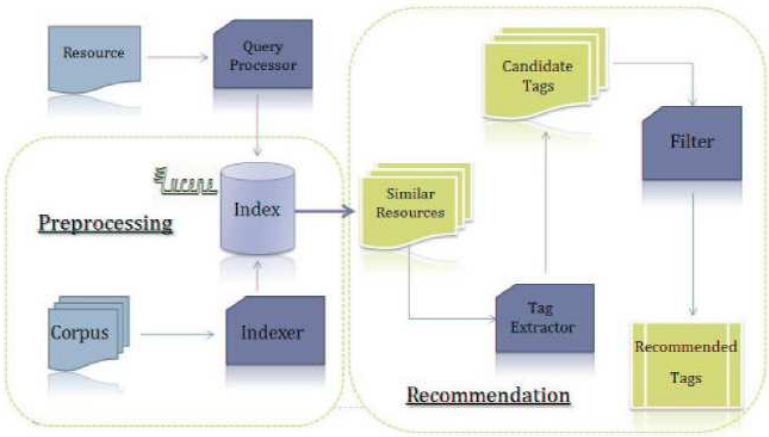

Figure 1: Architecture of STaR

Football Club (Inter.it). The *PersonalIndex*, instead, could return another online newspaper (Tuttosport.com).

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.

Figure 2 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 those not matching specific conditions, such as a threshold for the relevance score computed by the Tag Extractor. Obviously, the value of the threshold and the maximum number of tags to be recommended are strictly dependent from the training data. In the example in Figure 2, setting a threshold $\gamma = 0.20$, the system would suggest the tags *sport* and *newspaper*.

4. EXPERIMENTAL EVALUATION

The goal of experimental session was to tune the system parameters in order to obtain the best effectiveness of the tag recommender. We exploited a large dataset gathered from Bibsonomy.

4.1 Description of the dataset

The dataset used for the experimental evaluation 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.

4.2 Experimental Session

Firstly, we tried to evaluate the influence of different Lucene scoring functions on the performance of STaR. We randomly chose 10,000 resources from the dataset and we compared the results returned exploiting two different scoring functions (the Lucene original one and the BM25) in order to find the best one. 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 a low improvement by adopting BM25 with respect to the Lucene original similarity function. We can note that BM25 improved the recall of bookmarks (+ 6,95%) and BibTeX entries (+1,46%).

Next, using the BM25 as scoring function, we tried 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 *Social-*

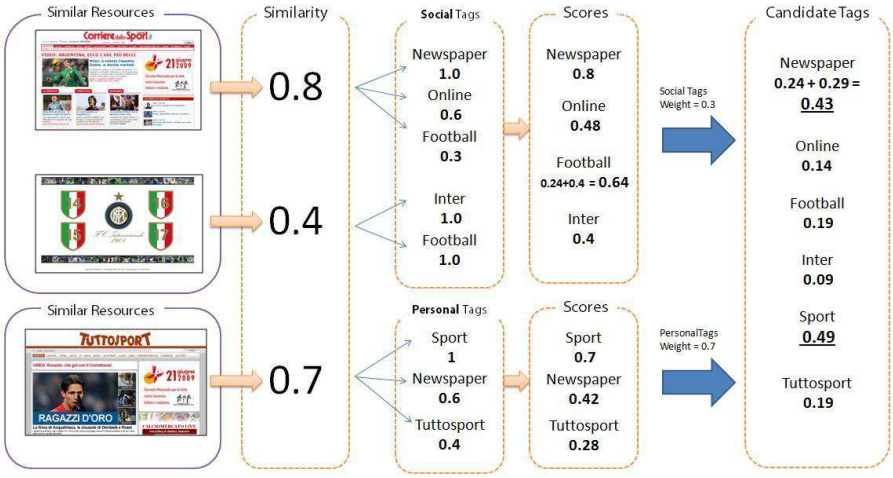

Figure 2: Description of the process performed by the Tag Extractor

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

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

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

TagWeight parameters;

- the threshold γ to establish whether a tag is relevant;
- which fields of the target resource use to compose the query.

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). A 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 a 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. Results are presented in Table 2 and Table 3.

Analyzing the results, 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

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

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 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

Nowadays, collaborative tagging systems are powerful tools but they are affected from some drawbacks since the complete tag space is too noisy to be exploited for retrieval and filtering tasks. In this paper we presented STaR, a social tag recommender system. The idea behind our work was to discover similarity among resources exploiting a state-of-the-art IR-model called BM25. 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).

This approach has a main drawback, since it cannot suggest any tags when the set of similar items returned by Lucene is empty. We are planning to extend the system in order to extract significant keywords from the textual content associated to a resource (title, description, etc.) that has no similar items, maybe exploiting structured data or domain ontologies.

Furthermore, since tags usually suffer of typical Information Retrieval problem (polysemy, etc.) we will try to establish whether the integration of Word Sense Disambiguation algorithms or a semantic representation of documents could improve the performance of the recommender.

Anyhow, our approach resulted promising compared with already existing and state of the art approaches for tag recommendation. Indeed, our work classified in 6th position in the final results of the ECML-PKDD 2009 Discovery Challenge (id: 29723)[4]

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] 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.

[4] 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.

[5] C. Musto, F. Narducci, M. de Gemmis, P. Lops, and G. Semeraro. STaR: a Social Tag Recommender System. In Folke Eisterlehner, Andreas Hotho, and Robert Jäschke, editors, *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497 of *CEUR Workshop Proceedings*, September 7 2009.

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

[7] 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.

[8] 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.

[4] http://www.kde.cs.uni-kassel.de/ws/dc09/results