

STaR: a Social Tag Recommender System

Cataldo Musto, Fedelucio Narducci, Marco de Gemmis,
Pasquale Lops, and Giovanni Semeraro

Department of Computer Science, University of Bari “Aldo Moro”, Italy
{musto,narducci,degemmis,lops,semeraro}@di.uniba.it

Abstract. The continuous growth of collaborative platforms we are recently witnessing made possible the passage from an ‘elitary’ Web, written by few and read by many, towards the so-called Web 2.0, a more ‘user-centric’ vision, where users become active contributors in Web dynamics. In this context, collaborative tagging systems are rapidly emerging: in these platforms users can annotate resources they like with freely chosen keyword (called tags) in order to make retrieval of information and serendipitous browsing more and more easier. 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. The goal of these systems (called tag recommenders) is to suggest a set of relevant keywords for the resources to be annotated by exploiting different approaches. In this paper we present a tag recommender developed for the ECML-PKDD 2009 Discovery Challenge. Our approach is based 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.

Key words: Recommender Systems, Web 2.0, Collaborative Tagging Systems, Folksonomies

1 Introduction

The coming of Web 2.0 has changed the role of Internet users and the shape of services offered by the World Wide Web. Since web sites tend to be more interactive and user-centric than in the past, users are shifting from passive consumers of information to active producers. By using Web 2.0 applications, users are able to easily publish content such as photos, videos, political opinions, reviews, so they are identified as *Web prosumers*: *producers* + *consumers* of knowledge. One of the forms of user-generated content (UGC) that has drawn more attention from the research community is *tagging*, which is the act of annotating

resources of interests with free keywords, called *tags*, in order to help users in organizing, browsing and searching resources through the building of a socially-constructed classification schema, called *folksonomy* [18]. In contrast to systems where information about resources is only provided by a small set of experts, collaborative tagging systems take into account the way individuals conceive the information contained in a resource [19]. Well-known example of platforms that embed tagging activity are Flickr¹ to share photos, YouTube² to share videos, Del.icio.us³ to share bookmarks, Last.fm⁴ to share music listening habits and Bibsonomy⁵ to share bookmarks and lists of literature. Although these systems provide heterogeneous contents, they have a common core: once a user is logged in, she can post a new resource and choose some significant keywords to identify it. Besides, users can label resources previously posted from other users. This phenomenon represents a very important opportunity to categorize the resources on the web, otherwise hardly feasible. The act of tagging resources from different users is the social aspect of this activity; in this way tags create a connection among users and items. Users that label the same resource by using the same tags could have similar tastes and items labeled with the same tags could have common characteristics.

Many would argue that the power of tagging lies in the ability for people to freely determine the appropriate tags for a resource without having to rely on a predefined lexicon or hierarchy [11]. Indeed, folksonomies are fully free and reflect the user mind, but they suffer of the same problems of unchecked vocabulary. Golder et. al. [5] identified three major problems with current 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 *turkey* could indicate a news taken from an online newspaper about politics or a recipe for Thanksgiving' Day. When multiple tags share a single meaning we refer to it as synonymy. In collaborative tagging systems we can have simple morphological variations (for example we can find 'blog', 'blogs', 'web log', to identify a common blog) but also semantic similarity (like resources tagged with 'arts' versus 'cultural heritage'). The third problem, called level variations, refers to the phenomenon of tagging at different level of abstraction. Some people can annotate a web page containing a recipe for roast turkey with the tag 'roast-turkey' but also with a simple 'recipe'.

In order to avoid these problems, in the last years many tools have been developed to facilitate the user in the task of tagging and to aid the tag convergence [4]: these systems are know as tag recommenders. When a user posts a resource in a Web 2.0 platform, a tag recommender suggests some significant keywords to label the item following some criteria to filter out the noise from the complete tag space.

¹ <http://www.flickr.com>

² <http://www.youtube.com>

³ <http://delicious.com/>

⁴ <http://www.last.fm/>

⁵ <http://www.bibsonomy.org/>

This paper presents STaR (Social Tag Recommender system), a tag recommender system developed for the ECML-PKDD 2009 Discovery Challenge. The idea behind our work is that folksonomies create connections among users and items, so we tried to point out two concepts:

- Resources with similar content could be annotated with similar tags;
- A tag recommender needs to take into account the previous tagging activity of users, by weighting more tags already used to annotate similar resources.

In this work we identify two main aspects in the tag recommendation task: firstly, each user has a typical manner to label resources (for example using personal tags such as ‘beautiful’, ‘ugly’, ‘pleasant’, etc. which are not connected to the content of the item, or simply tagging using general tags like ‘politics’, ‘sport’, etc.); next, similar resources usually share common tags: when a user posts a resource r on the platform, our system takes into account how she (if she is already stored in the system) and the entire community previously tagged resources similar to r in order to suggest relevant tags. Next, we develop this model and we tested it on a dataset extracted from BibSonomy.

The paper is organized as follows. Section 2 analyzes related work. The general problem of tag recommendation is introduced in Section 3. Section 4 explains the architecture of the system and how the recommendation approach is implemented. The experimental section carried out is described in Section 5.1, while conclusions and future works are drawn in last section.

2 Related Work

Previous work in the tag recommendation area can be broadly divided into three classes: *content-based*, *collaborative* and *graph-based* approaches.

In the content-based approach, a system exploits some textual source with Information Retrieval-related techniques [1] in order to extract relevant unigrams or bigrams from the text. Brooks et. al [3], for example, develop a tag recommender system that automatically suggests tags for a blog post extracting the top three terms exploiting TF/IDF scoring [14]. The system presented by Lee and Chun [8] recommends tags retrieved from the content of a blog using artificial neural networks. The network is trained based on statistical information about word frequencies and lexical information about word semantics extracted from WordNet. The collaborative approach for tag recommendation, instead, presents some analogies with collaborative filtering methods [2]. In the model proposed by Mishne and implemented in AutoTag [12], the system suggests tags based on the other tags associated with similar posts in a given collection. The recommendation process is performed in three steps: first, the tool finds similar posts and extracts their tags. All the tags are then merged, building a general folksonomy that is filtered and reranked. The top-ranked tags are suggested to the user, who selects the most appropriate ones to attach to the post. TagAssist [16] improves the AutoTags’ approach performing a lossless compression over existing tag data. It finds similar blog posts and suggests a subset of the associated tag through a

Tag Suggestion Engine (TSE) which leverages previously tagged posts providing appropriate suggestions for new content. In [10] the tag recommendations task is performed through a user-based collaborative filtering approach. The method seems to produce good results when applied on the user-tag matrix, so they show that users with a similar tag vocabulary tend to tag alike. The problem of tag recommendation through graph-based approaches has been firstly addressed by Jäschke et al. in [7]. They compared some recommendation techniques including collaborative filtering, PageRank and FolkRank. The key idea behind FolkRank algorithm is that a resource which is tagged by important tags from important users becomes important itself. The same concept holds for tags and users, thus the approach uses a graph whose vertices mutually reinforce themselves by spreading their weights. The evaluation showed that FolkRank outperforms other approaches. Schmitz et al. [15] proposed association rule mining as a technique that might be useful in the tag recommendation process. In literature we can find also some hybrid methods integrating two or more approaches (mainly, content and collaborative ones) in order to reduce their typical drawbacks and point out their qualities. Heymann et. al [6] present a tag recommender that exploits at the same time social knowledge and textual sources. They suggest tags based on page text, anchor text, surrounding hosts, adding tags used by others users to label the URL. The effectiveness of this approach is also confirmed by the use of a large dataset crawled from del.icio.us for the experimental evaluation. A hybrid approach is also proposed by Lipczak in [9]. Firstly, the system extracts tags from the title of the resource. Afterwards, based on an analysis of co-occurrences, the set of candidate tags is expanded adding also tags that usually co-occur with terms in the title. Finally, tags are filtered and reranked exploiting the information stored in a so-called "personomy", the set of the tags previously used by the user.

Finally, in [17] the authors proposed a model based on both textual content and tags associated with the resource. They introduce the concept of *conflated tags* to indicate a set of related tag (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. They win the previous edition of the Tag Recommendation Challenge.

3 Description of the Task

STaR has been designed to participate at the ECML-PKDD 2009 Discovery Challenge⁶. In this section we will firstly introduce a formal model for recommendation in folksonomies, then we will analyze the specific requirements of the task proposed for the Challenge.

⁶ <http://www.kde.cs.uni-kassel.de/ws/dc09>

3.1 Recommendation in Folksonomies

A collaborative tagging system is a platform composed of users, resources and tags that allows users to freely assign tags to resources. 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 $TAS: U \times R \rightarrow T$. The tag recommendation task for a given user $u \in U$ and a resource $r \in R$ can be finally described as the generation of a set of tags $TAS(u, r) \subseteq T$ 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.

3.2 Description of the ECML-PKDD 2009 Discovery Challenge

The 2009 edition of the Discovery Challenge consists of three recommendation tasks in the area of social bookmarking. We compete for the first task, content-based tag recommendation, whose goal is to exploit content-based recommendation approaches in order to provide a relevant set of tags to the user when she submits a new item (Bookmark or BibTeX entry) into Bibsonomy.

The organizers make available a training set with some examples of tag assignment: the dataset 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).

Each candidate recommender is evaluated by comparing the real tags (namely, the tags a user adopts to annotate an unseen resource) with the suggested ones. The accuracy is finally computed using classical IR metrics, such as Precision, Recall and F1-Measure (Section 5.1).

By analyzing the aforementioned requirements, we designed STaR thinking at a *prediction task* rather than a *recommendation* one. Consequently, we will try to emphasize the previous tagging activity of the user, also looking for connections and patterns among resources. All these decisions will be thoroughly analyzed in the next section describing the architecture of STaR.

4 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 [16] or AutoTag [12]. Although we agree with the idea that resources with similar content could be annotated with similar tags, in our opinion Mishne’s approach presents two important drawbacks:

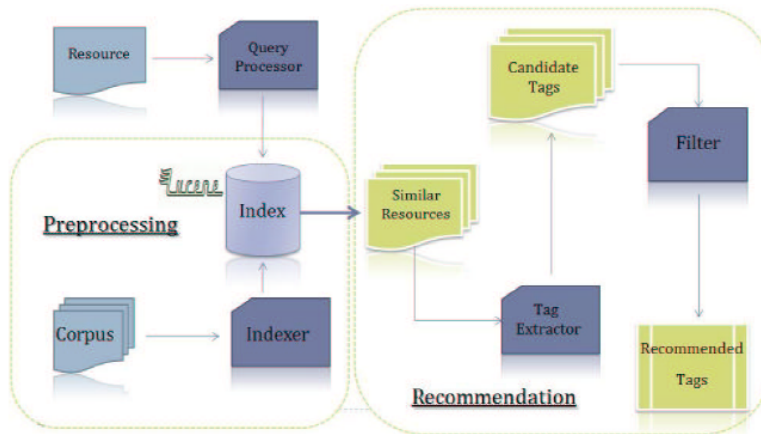


Fig. 1. Architecture of STaR

1. The tag reranking 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 based on the analysis of similar resources capable also of weighting more the tags already selected by the user during her previous tagging activity. 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.

4.1 Indexing of Resources

Given a collection of resources (*corpus*), a preprocessing step is performed by the *Indexer* module, which exploits Apache Lucene⁷ to perform the indexing step. As regards *bookmarks* we indexed the title of the web page and the extended description provided by users. For the *BibTeX* entries we indexed 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 her previously tagged resources and an index for the whole community (*Social Index*) storing the information about all the resources previously tagged by the community.

⁷ <http://lucene.apache.org>

Following the definitions presented in Section 3.1, given a user $u \in U$ we define *PersonalIndex*(u) as:

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

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:

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

4.2 Retrieving of Similar Resources

At the end of the preprocessing step STaR is able to take into account users requests. Every user interacts with STaR by providing information about a resource to be tagged. In the Query Processing step the system acquires data about the user (her language, the tags she uses more, the number of tags she usually uses to annotate resources, etc.) before processing (through the elimination of not useful characters and punctuation) and submitting the query against the *SocialIndex* stored in Lucene. If the user is recognized by the system since it has previously tagged some other resources, the same query is submitted against her own *PersonalIndex*, as well. We used as query the title of the web page (for bookmarks) or the title of the publication (for BibTeX entries). In order to improve the performances of the Lucene Querying Engine we replaced the original Lucene Scoring function with an Okapi BM25 implementation⁸. BM25 is nowadays considered as one of the state-of-the art retrieval models by the IR community [13].

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:

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

where $n_{t_i}^r$ represents the occurrences of the term t_i in the document d , $length_r$ is the length of the resource r and $avgLength_r$ is 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:

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

⁸ <http://nlp.uned.es/~jperezi/Lucene-BM25/>

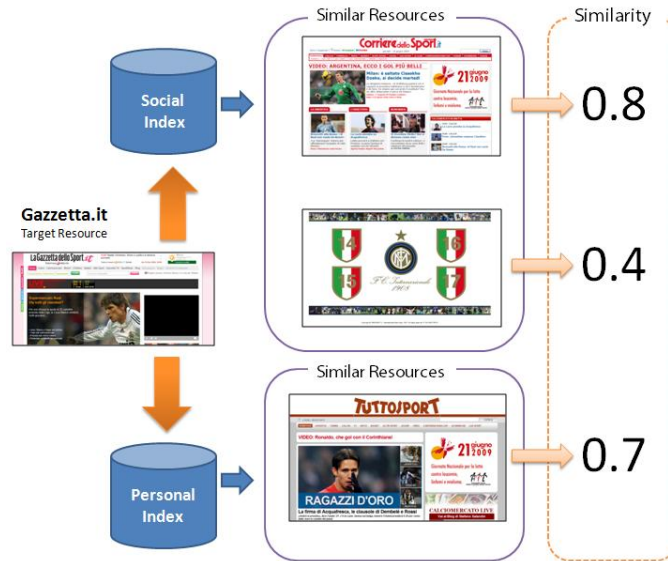


Fig. 2. Retrieving of Similar Resources

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:

$$PersonalRes(u, q) = \{r \in PersonalIndex(u) | sim(q, r) \geq \beta\} \quad (5)$$

$$SocialRes(q) = \{r \in SocialIndex | sim(q, r) \geq \beta\} \quad (6)$$

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

4.3 Extraction of Candidate Tags

In the next step the *Tag Extractor* gets the most similar resources returned by the Apache Lucene engine and produces the set of candidate tags to be suggested, by computing for each tag 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 users' previously used tags.

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:

$$candTags_p(u, q) = \{t \in T | t = TAS(u, r) \wedge r \in PersonalRes(u, q)\} \quad (7)$$

$$candTags_s(q) = \{t \in T | t = TAS(u, r) \wedge r \in SocialRes(q) \wedge u \in U\} \quad (8)$$

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

$$rel_p(t, u, q) = \frac{\sum_{r \in PersonalRes(u, q)} n_r^t * sim(r, q)}{n^t} \quad (9)$$

$$rel_s(t, q) = \frac{\sum_{r \in SocialRes(q)} n_r^t * sim(r, q)}{n^t} \quad (10)$$

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:

$$candTags(u, q) = candTags_p(u, q) \cup candTags_s(q) \quad (11)$$

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

$$rel(t, q) = \alpha * rel_p(t, q) + (1 - \alpha) * rel_s(t, q) \quad (12)$$

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.

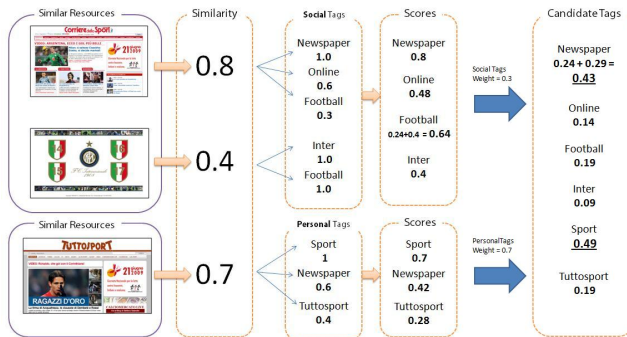


Fig. 3. Description of the process performed by the Tag Extractor

4.4 Tag Recommendation

The *Tag Extractor* produces the set of the *Candidate Tags*, a ranked set of tags with their relevance scores. This set is exploited by the *Filter*, a component which performs the last step of the recommendation task, that is removing those tags not matching specific conditions: we fix a threshold for the relevance score between 0.20 to 0.25 and we return at most 5 tags. These parameters are 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 $recommendation(u, q)$ defined as follows:

$$recommendation(u, q) = \{t \in candTags(u, q) | rel(t, q) > \gamma\} \quad (13)$$

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

5 Experimental Evaluations

5.1 Experimental Session

In this experiment we measure the performance of STaR in the Task 1 of the ECML-PKDD 2009 Discovery Challenge. This experimental evaluation was carried out according to the instructions provided from the organizers of the Challenge 2009. The test set was released 48 hours before the end of the competition. Every participant uploaded a file containing the tag predictions, and for each post only five tags were considered. F1-Measure was used to evaluate the accuracy of recommendations, thus for each post Precision and Recall were computed by comparing the recommended tags with the true tags assigned by the users. The case of tags was ignored and all characters which are neither numbers nor letters were removed. Results are presented in Table 1.

Table 1. Results of the ECML-PKDD 2009 Discovery Challenge

#Tag	Precision	Recall	F1
1	19.51	6.89	10.19
2	16.34	10.10	12.53
3	14.55	12.16	13.25
4	13.56	13.53	13.55
5	13.56	13.53	13.55

STaR finished the ECML-PKDD Discovery Challenge 2009 with an overall F-measure of 13.55. As showed in the table above, exploiting only the first recommended tag the system reaches almost 20% in precision. The value of the recall increases with the number of recommended tags reaching the 13.5% in the fourth and fifth tag. In the future we will perform a more in-depth study in order to compare the predictive accuracy of STaR with different configurations of parameters.

6 Conclusions and Future Work

In this paper we presented STaR, a tag recommender designed and implemented to participate to the ECML-PKDD 2009 Discovery Challenge. 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).

In the future we will implement a methodology to suggest tags when the set of similar items returned by Lucene is empty. The system should be able 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. Another issue to investigate 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, textual content suffers from syntactic problems like polysemy (a keyword with two or more meanings) and synonymy (two or more keywords with the same meaning). These problems hurt the performance of the recommender. We will try to establish if a semantic document indexing could improve the performance of the recommender.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

2. D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *Proceeding of the 15th International Conference on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.
3. 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.
4. 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.
5. S. Golder and B. A. Huberman. The Structure of Collaborative Tagging Systems. *Journal of Information Science*, 32(2):198–208, 2006.
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 - Adaptivität (LWA 2007)*, pages 13–20, September 2007.
8. Sigma On Kee Lee and Andy Hon Wai Chun. Automatic tag recommendation for the web 2.0 blogosphere using collaborative tagging and hybrid semantic structures. In *ACOS'07: Proceedings of the 6th Conference on WSEAS International Conference on Applied Computer Science*, pages 88–93, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).
9. M. Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 84–95, 2008.
10. Leandro B. Marinho and Lars Schmidt-Thieme. Collaborative tag recommendations. pages 533–540. 2008.
11. Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>, December 2004.
12. Gilad 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.
13. Stephen E. Robertson, Steve Walker, Micheline H. Beaulieu, Aaron Gull, and Marianna Lau. Okapi at trec. In *Text REtrieval Conference*, pages 21–30, 1992.
14. G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.
15. Christoph Schmitz, Andreas Hotho, Robert Jschke, and Gerd Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. ?iberna, 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.
16. Sanjay Sood, Sara Owsley, Kristian Hammond, and Larry Birnbaum. TagAssist: Automatic Tag Suggestion for Blog Posts. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.
17. 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.

18. Thomas Vander Wal. Folksonomy coinage and definition. Website, Februar 2007. <http://vanderwal.net/folksonomy.html>.
19. Harris Wu, Mohammad Zubair, and Kurt 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.