Combining Collaborative and Content-Based Techniques for Tag Recommendation

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

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

Abstract. The explosion of collaborative platforms we are recently witnessing, such as social networks, or video and photo sharing sites, radically changed the Web dynamics and the way people use and organize information. The use of tags, keywords freely chosen by users for annotating resources, offers a new way for organizing and retrieving web resources that closely reflects the users' mental model and also allows the use of evolving vocabularies. However, since tags are handled in a purely syntactical way, the annotations provided by users generate a very sparse and noisy tag space that limits the effectiveness of tag-based approaches for complex tasks. Consequently, systems called tag recommenders recently emerged, with the purpose of speeding up the so-called tag convergence, providing users with the most suitable tags for the resource to be annotated.

This paper presents a tag recommender system called STaR (Social Tag Recommender), which extends the social approach presented in a previous work [14] with a content-based approach able to extract tags directly from the textual content of HTML pages.

Results of experiments carried out on a large dataset gathered from Bibsonomy, show that the use of content-based techniques improves the predictive accuracy of the tag recommender.

Keywords: Recommender Systems, Web 2.0, Collaborative Tagging Systems, Folksonomies, Tag Recommender.

1 Introduction

The recent explosion of collaborative platforms radically changed the nature of the services offered by the Web. For many years Internet users have been relegated to a passive role in the Web dynamics: they were only viewers of the information created by other users (e.g. web site administrators) and they were not able to contribute in any way to enrich the available content. Nowadays this strong dichotomy has been replaced by a new and more democratic vision, in which users are more and more involved, because they are able to:

1. produce new content (the so-called User Generated Content);
2. enrich already available content with novel metadata.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 13–23, 2010.

In platforms like YouTube, Flickr or Wikipedia users contribute in the first way, by sharing resources (new videos, photos, . . .) that other users can enjoy, while in the collaborative tagging systems, the contribution of the community falls into the second category.

The idea behind the concept of tagging is simple: a user enjoys a resource (an image, a web site, etc.) and, according to her mental model, identifies those terms that better describe the information conveyed by that resource. The same resource can be annotated by several users: some of them will reuse the tags already assigned to that resource, while some others will adopt new tags. This process allows building a socially-constructed classification schema, called folksonomy [18]. The inceptive idea behind folksonomies is that the more a tag is used by the community to annotate the target resource, the more is the likelihood the tag correctly describes its content.

Recently the use of folksonomies gained more attention because of their simplicity: using tags, users can freely model the information without the constraints of a predefined lexicon or hierarchy [12]. However, the simplicity of the approach has also an important drawback: the information managed by folksonomies is modeled in a simple syntactical way. Therefore, as stated by Golder et. al. [8], collaborative tagging systems suffer from the classic problems of Information Retrieval (IR) systems like polysemy, synonymy and level variation.

The *polysemy* refers to situations where tags can have multiple meanings: for example a resource tagged with the term *turkey* could indicate a news in an online newspaper about politics or a recipe for Thanksgiving Day.

When multiple tags share the same meaning we refer to it as *synonymy.* In collaborative tagging systems we can also have simple morphological variations, for example we can find *blog*, *blogs*, *web log*, to identify a common blog, but also tags semantically similar such as resources tagged with *arts* versus *cultural heritage.*

Finally, the *level variation* problem 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 *roastturkey*, but also with a simple *recipe.*

These drawbacks hinder the use of folksonomies for tasks more complex than the simple browsing of resources. In order to avoid these problems, in the last years many tools have been developed to facilitate the user in the task of tagging, by also speeding up the tag convergence [6]: these systems are known 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);
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.

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.

We applied STaR to the task of recommending tags for bookmarks. In this work we enriched the approach we proposed in [14] by integrating some heuristics for extracting tags from the HTML content of web pages to be annotated. This could allow producing recommendations even when no similar bookmarks are available, and the process of extracting tags from similar resources inevitably fails.

The paper is organized as follows. Section 2 analyzes related work, while Section 3 depicts the general system architecture and the recommendation approach implemented in STaR. Details about the experimental evaluation are given in Section 4. Finally, conclusions and future works are drawn in the last section.

2 Related Work

Usually tag recommenders are broadly divided into three classes: *content-based*, *collaborative*, and *graph-based* approaches.

A content-based tag recommender exploits textual content by adopting Information Retrieval-related techniques [1] in order to extract relevant terms (unigrams or bigrams) to label a resource. Brooks et. al [5], for example, developed a tag recommender system that automatically suggests tags for a blog post by extracting the top three terms exploiting the TF/IDF scoring [15]. Another work exploiting the TF/IDF measure is KEA [19]. This system applies a supervised Bayesian classifier able to extract relevant phrases from textual content.

Other systems exploit ontologies for recommending tags: in [2], terms are extracted from the document and subsequently, surfing the ontology, more abstract and conceptual tags are suggested.

The collaborative approach for tag recommendation, instead, presents some analogies with collaborative filtering methods [4]. These systems compute the relevance of a tag to be suggested by exploiting the community behavior. In the model proposed by Mishne and implemented in AutoTag [13], 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 re-ranked. The top-ranked tags are suggested to the user, who selects the most appropriate ones to attach to the post. In [7] the authors propose an adaptation of the classical K-nearest neighbor approach to create a set of recommended tags. The neighbors are defined looking for users tagging the same resources with the same tags. In this way tags used by similar users are boosted in the set of recommended tags.

The problem of tag recommendation through graph-based approaches has been firstly addressed by Jäschke et al. in [9]. 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 vertexes mutually reinforce themselves by spreading their weights. The evaluation showed that FolkRank outperforms other approaches. Schmitz et al. [16] 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. In [11] the authors presents a tag recommender that suggests tags based on the resource content, resource related tags and user profile tags. The content exploited by the recommender is the title of the resource (and the URL, if available). Then, the system extends the set of words extracted from the title with tags related to the title as well as with tags occurring in the user and resource profile. The user profile is composed by all the tags used by that specific user to label resources. Instead, the resource profile groups tags assigned to a resource by other users. Another similar approach is presented in [10] where the authors exploit three kinds of information sources: the description of the resources, their folksonomies, and the tags previously used by the same person. In order to remove inappropriate candidate tags, a filtering method and a weighting scheme for assigning different importance to sources are applied.

3 STaR: A Social Tag Recommender System

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 and a specific resource can be described as the generation of a set of tags 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 is a tag recommender system, developed at the University of Bari. The inceptive idea behind STaR is to improve the model implemented in systems like AutoTag [13]. Although we agree with the idea that resources with similar content could be annotated with similar tags, in our opinion the approach proposed in [13] presents three 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.
3. When there are not similar resources available for suggesting similar tags, the recommendation task fails.

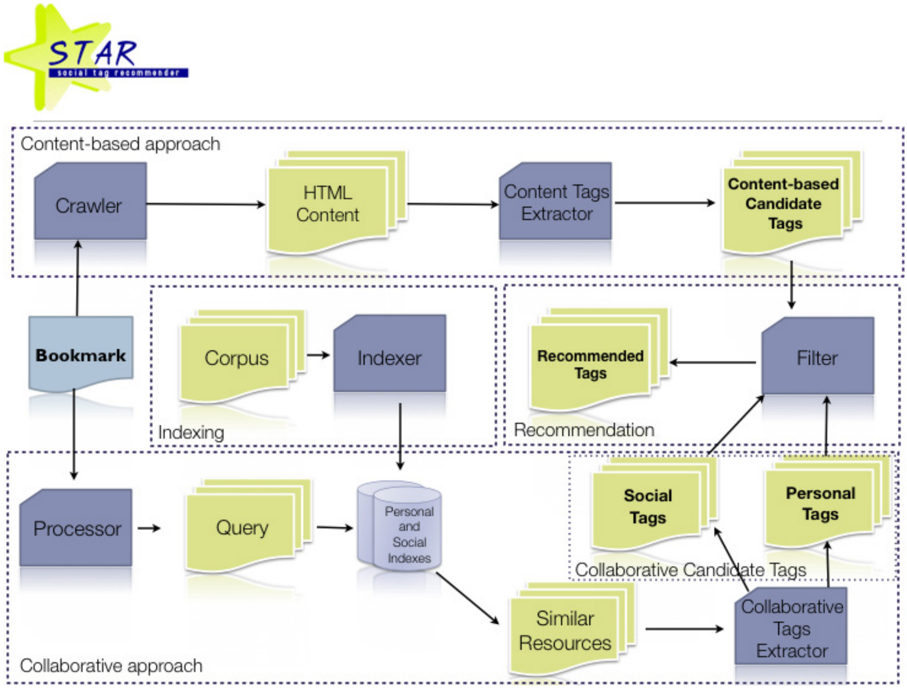

Fig. 1. Architecture of STaR

We will try to overcome these drawbacks by proposing an approach based on the analysis of similar resources, also capable of weighting more the tags already selected by the user during her previous tagging activity. Furthermore, we also integrated some heuristics to directly extract tags from the HTML content of a Web page.

Figure 1 shows the general architecture of STaR. In our system we integrated two different approaches to recommend tags: the content-based and the collaborative one. The collaborative approach needs a preprocessing step that, given a corpus of documents, produces a Social and a Personal indexes that are queried in order to produce a set of *Collaborative Candidate Tags*.

In the same way a set of *Content-based Candidate Tags* is produced by crawling the HTML content of the web page. Finally, the recommendation step merges the results obtained by the aforementioned approaches.

In the next section the recommendation model implemented in STaR is thoroughly analyzed.

3.1 Indexing Step

Given a collection of resources already annotated (tagged), called *corpus*, a preprocessing step is performed by the *Indexer* module, which exploits Apache

Lucene[1] to perform the indexing step. The corpus is composed by *bookmarks*, so we indexed the title of the web page, the description provided by users and the tags used to annotate it. We build an index for each user, called *Personal Index*, that stores the information on her previously tagged resources and an index for the whole community, called *Social Index*, that stores the information about all the resources previously tagged by the community.

3.2 Tag Extraction Step

STaR implements two different approaches to recommend tags for a new bookmark: a *collaborative* approach, and a *content-based* one.

- **Collaborative approach**
  After the Indexing step, STaR retrieves the most similar bookmarks (to that to be annotated) and collects the most relevant tags used by the community to annotate them.

  More specifically, given a user u and a resource r to be annotated, STaR returns the resources whose similarity with r is greater or equal than a threshold β whose value is dependent from the corpus used for the evaluation. To perform this task, the *Processor* gets data about both the user (the tags she uses more) and the resource (metadata like the title of the web page) and submits a query against the *Social Index*. If the user is recognized by the system, since it has previously tagged some other resources, the same query is submitted against her own *Personal Index*, as well. The *Collaborative Tags Extractor* builds a set of *Collaborative Candidate Tags* by extracting tags assigned to the most similar resources retrieved from the indexes. For each tag, a score is computed by weighting the similarity score returned by Apache Lucene with the normalized occurrence of the tag. This list is finally ranked and the top n tags are suggested to the user.

  In order to improve the performance of the Lucene Querying Engine the original Lucene Scoring function has been replaced with an Okapi BM25 implementation[2]. In [14] we showed that the BM25 integration improves the overall accuracy of STaR. Further details on the collaborative approach implemented in STaR are provided in [14].
- **Content-based approach**
  The collaborative approach fails when STaR does not retrieve any bookmark similar to the one the user should annotate.

  To tackle this problem, we implemented a content-based approach for extracting tags by analyzing the HTML source of a web page. The idea behind this strategy is to identify a set of candidate tags, and to choose the more promising ones by analyzing the content of the web page. Starting from the URL of the new bookmark posted by the user, the *Crawler* retrieves the web page and its HTML source. Next, the *Content Tags Extractor* extracts from the URL the string representing the domain name (for example from

[1] `http://lucene.apache.org`

[2] `http://nlp.uned.es/%7Ejperezi/Lucene-BM25/`

the URL *www.wikipedia.it*, we extract *wikipedia*). This represents the first element in the set of *Content-based Candidate Tags*. Afterwards, we extract other candidate tags by analyzing the content stored in the HTML tags $\langle title \rangle$ and $\langle meta \rangle$. As regards the tag $\langle meta \rangle$, we extract the value of the attributes *keywords* and *description*, because in our opinion they are the most significant. This list is filtered by deleting stopwords and verbs.

When the set of *Content-based Candidate Tags* is built, the algorithm assigns a score to each tag, according to several heuristics. First, the number of occurrences of each candidate tag in the HTML source is taken into account (values are normalized in a range between 0 and 1). Then, a different weight is assigned to each source according to the following values (this is a heuristics that needs to be evaluated):

- URL: 0.2
- TITLE: 0.3
- META KEYWORDS: 0.25
- META DESCRIPTION: 0.25

The score assigned to each tag is the weighted sum of the normalized occurrence of the tag in each source multiplied for the weight of the same source.
More formally, given $S = \{s_1, ..., s_n\}$ the set of sources, $Cand = \bigcup_{s_i \in S} Cand_s$, where $Cand_s$ is the set of candidate tags coming from the source s, the score of the tag t is computed by the following formula:

$$score(t) = \sum_{s_i \in S} w_{s_i} \cdot n_t \qquad (1)$$

where s_i identifies the source of the candidate tag, w_{s_i} is the weight of the source s_i, and n_t is the normalized occurrence of the tag t in the HTML source ($n_t = 0$ if the tag t does not occur in the source s_i).

3.3 Tag Recommendation Step

After the tag extraction step, two sets of tags, *Content-based Candidate Tags* and *Collaborative Candidate Tags* are available. The set *Collaborative Candidate Tags* contains the set *Social Tags* and *Personal Tags*.

The *Filter* component removes from each set those tags whose score is under a certain threshold, and decides which set to exploit to provide recommendations. The *Filter* can suggest just the tags belonging to a specific set, or it can implement a specific strategy to combine different sets.

The strategy implemented in this work is a *cascade* one: the *Filter* starts from a unique set of tags (*Social*, *Personal*, or *Content-based*) and, if that set is empty, it uses the others. For example, it starts with the *Personal Tags* set, but if the user is new in the community, that set is empty; thus, *Social Tags* set is used. If this set is empty too, for example because there are not similar resources to the one to be annotated, the last chance is to exploit tags extracted by the content-based approach.

The *Filter* can implement other strategies: for example a merge of the three sets of tags can be defined. Possible strategies will be investigated in future works.

4 Experimental Evaluation

We designed two different experimental sessions to evaluate the performance of STaR. In the first session we evaluated the predictive accuracy of the single sets of tags, *Social Tags*, *Personal Tags* and *Content-based Tags*, while in the second session we investigated whether the combination of these sets helps STaR to produce better recommendations.

4.1 Description of the Dataset

In the experiment we adopted the dataset used for the Content-based recommendation task of the ECML-PKDD 2009 Discovery Challenge[3]. We just used the *263,004 bookmark posts* submitted by 3,617 different users. For each of the 235,328 different URLs were also provided some textual metadata (such as the title of the resource, the description and so on).

We evaluated STaR by comparing the real tags (namely, the tags a user adopts to annotate a new resource) with the suggested ones. The accuracy was finally computed using classical Information Retrieval metrics, such as Precision (Pr), Recall (Re) and F1-Measure (F1) [17].

4.2 Experimental Session

In the first experiment we evaluated the predictive accuracy of the single sets of tags produced by the *Content-Based Tags Extractor* and the *Collaborative Tags Extractor*. In this experiment the threshold β was set equal to 0.20. Results are presented in Table 1.

Table 1. Predictive accuracy of the single sets of tags

Recommendation set	Pr	Re	F1
Personal Tags	11.20	8.34	9.58
Social Tags	13.80	11.28	12.41
Content-based Tags	14.36	17.90	15.90

The effectiveness of the collaborative approach (Personal Tags and Social Tags) is strictly dependent on the amount of the available resources already annotated. Since the approach based on Personal Tags builds the set of candidates tags on the ground of the resources the user annotated in the past, we observed a significant loss in recall with respect to the content-based approach (-9.56). In

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

the approach based on Social Tags the recommendation model analyzes the set of resources already tagged by all the users in the community, and this reduces the loss of recall with respect to the content-based approach (-6.62).

As regards the overall accuracy of the content-based approach in terms of F1, it outperforms the collaborative one (+3.49 for Social Tags and +6.32 for Personal Tags). This can be explained by observing that the content-based model is totally independent from the number of the resources already annotated in the corpus, and also because it does not suffer of the cold-start problem. The content-based method based on the extraction of tags from the HTML content of the web page obtained a Precision comparable to that of Social Tags, but with a significant improvement in recall (+6.62).

To sum up, the main outcome of the first experiment is that the content-based approach proposed in the paper is a valuable strategy for improving the performance of the tag recommender.

In the second experiment we investigated whether some strategies for combining collaborative and content-based approaches are useful to improve the predictive accuracy of STaR.

We run the content-based (respectively, the collaborative) tag extraction algorithm when the collaborative (respectively, the content-based) one is not able to return any tag. As regards the collaborative model, this happens when no similar documents are retrieved by querying the corpus containing previously annotated resources, while in the content-based model, this happens when the HTML source does not include any meta-tag or when the words contained in the set of *Content-based Candidate Tags* do not occur in the HTML source code. Results are presented in Table 2.

Table 2. Predictive accuracy of different strategies for combining collaborative and content-based approaches

Configuration	Pr	Re	F1
Personal+Social Tags *(baseline)*	14.28	16.53	15.32
Personal+Content-based Tags	13.64	15.51	14.52
Content-based+Personal Tags	17.79	15.07	16.32
Social+Content-based Tags	11.50	13.96	12.61
Personal+Social+Content-based Tags	14.51	16.68	15.52
Content-based+Personal+Social Tags	**19.21**	**16.12**	**17.53**

The Personal+Social Tags approach is used as baseline since it represents the configuration of STaR used to participate to the ECML-PKDD 2009 Discovery Challenge [14]. The first outcome of the experiment is that the combination of techniques for recommending tags is a valuable strategy. Indeed, the baseline outperforms the results obtained using just Personal Tags (F1: +5.74) or Social Tags (F1: +2.91). It is also worth to note that the order different techniques are used matters. The configuration of the experiment adopting the content-based tags first and the set of personal tags afterwards outperforms the configuration adopting personal tags first and content-based tags subsequently (F1: +1.8). This result is coherent with the last two results reported Table 2 (F1: +2.01). Another interesting result is that using configuration exploiting content-based tags as the first strategy gives the best results. This is in line with results report in Table 1, in which the best performance of STaR was obtained by adopting the set of content-based tags.

5 Conclusions and Future Work

In this paper we presented STaR, a tag recommender system integrating a social approach with a content-based one able to extract tags by analyzing the textual content of resources to be annotated. The inceptive 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. We also enriched this collaborative approach by extracting tags from HTML source code by adopting simple heuristics: we showed that this improves the predictive accuracy of the recommendation model.

We are planning to run several experiments related to the content-based extraction algorithm in order to tune the several systems parameters. Finally, we will investigate about other possible strategies for combining and merging tags coming from the different sources: for example a simple linear combination of the relevance scores could be adopted.

Furthermore, since tags usually suffer of typical Information Retrieval problem (polysemy, synonymy, etc.) we will try to establish if the integration of Word Sense Disambiguation [3] tools or a semantic representation of documents could improve the performance of the tag recommender.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)
2. Baruzzo, A., Dattolo, A., Pudota, N., Tasso, C.: Recommending new tags using domain-ontologies. In: Web Intelligence/IAT Workshops, pp. 409–412 (2009)
3. Basile, P., Degemmis, M., Gentile, A.L., Lops, P., Semeraro, G.: UNIBA: JIGSAW algorithm for Word Sense Disambiguation. In: Proceedings of the 4th ACL International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, June 23-24, 2007, pp. 398–401. Association for Computational Linguistics (2007)
4. Billsus, D., Pazzani, M.J.: Learning collaborative information filters. In: Proceeding of the 15th International Conference on Machine Learning, pp. 46–54. Morgan Kaufmann, San Francisco (1998)
5. Brooks, C.H., Montanez, N.: Improved annotation of the blogosphere via autotagging and hierarchical clustering. In: WWW '06: Proceedings of the 15th International Conference on World Wide Web, pp. 625–632. ACM, New York (2006)

6. Cattuto, C., Schmitz, C., Baldassarri, A., Servedio, V.D.P., Loreto, V., Hotho, A., Grahl, M., Stumme, G.: Network properties of folksonomies. AI Communications 20(4), 245–262 (2007)
7. Gemmell, J., Schimoler, T., Ramezani, M., Mobasher, B.: Adapting k-nearest neighbor for tag recommendation in folksonomies. In: ITWP (2009)
8. Golder, S., Huberman, B.A.: The Structure of Collaborative Tagging Systems. Journal of Information Science 32(2), 198–208 (2006)
9. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In: Hinneburg, A. (ed.) Workshop Proceedings of Lernen - Wissensentdeckung - Adaptivität (LWA 2007), September 2007, pp. 13–20 (2007)
10. Ju, S., Hwang, K.: A weighting scheme for tag recommendation in social bookmarking systems. In: Eisterlehner, F., Hotho, A., Jäschke, R. (eds.) ECML PKDD Discovery Challenge 2009 (DC'09), CEUR Workshop Proceedings, September 2009, vol. 497, pp. 109–118 (2009)
11. Lipczak, M., Hu, Y., Kollet, Y., Milios, E.: Tag sources for recommendation in collaborative tagging systems. In: Eisterlehner, F., Hotho, A., Jäschke, R. (eds.) ECML PKDD Discovery Challenge 2009 (DC'09), CEUR Workshop Proceedings, September 2009, vol. 497, pp. 157–172 (2009)
12. Mathes, A.: Folksonomies - cooperative classification and communication through shared metadata (December 2004), `http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html`
13. Mishne, G.: Autotag: a collaborative approach to automated tag assignment for weblog posts. In: WWW '06: Proceedings of the 15th International Conference on World Wide Web, pp. 953–954. ACM, New York (2006)
14. Musto, C., Narducci, F., de Gemmis, M., Lops, P., Semeraro, G.: Star: a social tag recommender system. In: Eisterlehner, F., Hotho, A., Jäschke, R. (eds.) ECML PKDD Discovery Challenge 2009 (DC'09), CEUR Workshop Proceedings, Bled, Slovenia, vol. 497, pp. 215–227 (2009)
15. Salton, G.: Automatic Text Processing. Addison-Wesley, Reading (1989)
16. Schmitz, C., Hotho, A., Jäschke, R., Stumme, G.: Mining association rules in folksonomies. In: Data Science and Classification (Proc. IFCS 2006 Conference), Studies in Classification, Data Analysis, and Knowledge Organization, July 2006, pp. 261–270. Springer, Heidelberg (2006)
17. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)
18. Vander Wal, T.: Folksonomy coinage and definition. Website (Februar 2007), `http://vanderwal.net/folksonomy.html`
19. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco (1999)