Finding Communities of Practice from User Profiles Based On Folksonomies

Jörg Diederich and Tereza Iofciu

L3S Research Center, Leibniz Universität Hannover
Expo Plaza 1, 30539 Hannover, Germany
Email: {diederich, iofciu}@l3s.de

Abstract. User profiles can be used to identify persons inside a community with similar interests. Folksonomy systems allow users to individually tag the objects of a common set (e.g., web pages). In this paper, we propose to create user profiles from the data available in such folksonomy systems by letting users specify the most relevant objects in the system. Instead of using the objects directly to represent the user profile, we propose to use the tags associated with the specified objects to build the user profile. We have designed a prototype for the research domain to use such *tag-based profiles* in finding persons with similar interests. The combination of tag-based profiles with standard recommender system technology has resulted in a new kind of recommender system to recommend related publications, keywords, and persons. Especially the latter is useful to find persons to potentially cooperate with and to monitor the community to be able to enhance a user's current Community of Practice.

1 Introduction

For people in a community (such as professors and students in the research community), a well-defined profile expressing their current interests is highly valuable. As one main application, such profiles can help to find persons who work on related topics and, thus, help to facilitate cooperation within the community.

Two steps are necessary to create user profiles:

1. Determine the user profile schema, i.e., how the user profile should look like.
2. Determine how to populate the user profiles with actual data for particular users.

Both steps are interrelated: In general, the higher the accuracy of the user profile is, the more data the profile schema comprises, and a large schema in general leads to more complex handling and maintenance of the profiles. Especially the problem of populating user profiles with actual and accurate data is difficult to solve for large profiles as accurate data mostly is based on human inspection.

In this paper, we propose to use tagged corpora of objects to create user profiles in domains, where such folksonomies are available. The basic idea is to let people create their profiles by specifying the most relevant objects in the folksonomy. Afterwards, this *intermediate profile* comprising the objects is translated into the tag domain, assuming that the manually specified tags describe the objects with a high accuracy. Hence, the

E. Tomadaki and P. Scott (Eds.): Innovative Approaches for Learning and Knowledge Sharing,
EC-TEL 2006 Workshops Proceedings, ISSN 1613-0073, p. 288-297, 2006.

representation of the *final user profile* is based on the tags of the most relevant objects. This has the advantage that users only have to specify comparatively few objects to generate a reasonably large user profile. Furthermore, it is easier to find related user profiles as tags are typically shared by several objects.

We apply our approach to the domain of digital libraries, using a subset of the DBLP data set as object corpus, which has been enhanced with 'tags', e.g., the keywords that were manually specified by the authors of the publications. The resulting user profiles, generated by our prototypical *TBProfile system*, are represented by keyword vectors and are exported in RDF (as already proposed in the eLearning domain [5]), so they can be reused in other domains with similar tags. The TBProfile system uses standard recommender system technology on these profiles to recommend other publications, other relevant keywords (for refining the user profile), and finally other relevant persons. These persons, being relevant for the user, are potential candidates to collaborate with and, thus, to be added to the user's Community of Practice.

This paper is organized as follows: The related work is given in Section 2. In Section 3 we describe our approach to creating and maintaining user profiles and present our experimental setup. Section 4 describes how to provide users with relevant recommendations based on these user profiles and how to build communities of practice. We conclude and outline future research directions in Section 5.

2 Related Work

There are different approaches to extracting user profiles from users' past activities and using them for discovering and analyzing communities. In [4], the similarity between peers in social collaboration networks is used to improve search in a peer-to-peer network. The similarity is computed based on publications and their references. The user profile is build based on the publications the user has stored on her desktop. This approach is too broad as the documents a user stores are usually not focused enough. The system takes into account all publications found, including ones dealing with topics the user may no longer have interest in or that the user has stored without even reading them or working on the topic.

Middleton et al. [9] present a recommender system for online academic publications where user profiling is done based on a research paper topic ontology. The system monitors what research papers a group of person has downloaded from the web and stores them on a server. For all downloaded research papers, terms are extracted from the full text using standard information retrieval techniques to be able to represent the paper with term vectors. The system uses different classifiers to assign topics to the papers. User profiles are automatically built based on the vector-representation of those research papers, downloaded by a particular person in the monitored group of persons, and can be refined based on relevance feedback. Finally, the system gives recommendations for each user based on the user's profile. While an automatic update of the profile based on actual browsing of papers (similar to other publication recommender systems [1, 11]) can reduce the efforts for creating and maintaining user profiles, this is in contrast to the issue that user profiles are typically rather stable over time, while the 'browsing task' is often focused on a short-term goal (e.g., help a colleague to find

something or explore a topic which finally turns out not to be interesting). Hence, not all browsed documents are relevant to the user, even if we take into account the time spent on the respective document. Also, we would like to limit the collection of explicit relevance feedback which can create quite a workload for the user. Furthermore, the approach is pretty intrusive as it requires the monitoring of the browsing behavior of a group of persons. In contrast, our approach is based on publicly available information about objects and manually-assigned tags of objects. As manually assigned tags are assumed to be highly accurate, our approach does not suffer from the inaccuracy of an automatic classification system.

Existing systems to recommend publications in the domain of research are mainly keyword-based search engines (e.g., google scholar, ACM digital library etc.). They are mainly intended to fulfill short-term search objectives (find a paper with a specific title, find the paper for a specific author etc.). However, some papers are difficult to find based on keywords only, especially if a research domain is already well known. Furthermore, once a researcher has written a paper, she might turn to a different topic within her research interests, but still would like to be informed about the development in some of the topics, she has previously worked on. Hence, a recommender system for research papers [8] based on a long-term user profile is highly desirable. While the issue of user profiles has been found to be highly relevant for recommender systems [10], it has not been addressed sufficiently in the literature, and there are no existing systems which share the user profiles they are using to take advantage of the distributed knowledge about the users. This gap is intended to be filled by our TBProfile prototype.

3 TBProfile: A Tag-Based User Profile Generator

This section presents our approach to creating and maintaining user profiles. The basic idea is to relate a user with a set of tagged objects and store them in an intermediate user profile. The final representation of the user profile is based on the tags associated with the objects. An example set of objects (publications from the Semantic Web domain) forming an intermediate user profile is shown in Table 1.

Publication title	Tags (Keywords)
Magpie: supporting browsing and navigation on the semantic web	named entity recognition (NER), semantic web, semantic web services, ...
Bootstrapping ontology alignment methods with APFEL	alignment, mapping, ontology, ...
Swoogle: a search and metadata engine for the semantic web	rank, search, semantic web, ...

Table 1. Example: Intermediate user profile comprising a set of tagged publications

A user having selected only these three publications will be described by the final user profile shown in Table 2. Using the tags in the user profile has several advantages:

User	...	NER	Semantic Web	SW Services	Alignment	Mapping	ontology	rank	search ...	
A	...	1	2	1	1	1	1	1	1	...

Table 2. Example for the final representation of a user profile

- A more accurate description of the user's interests based on the content of the selected objects.
- A denser population of the user profile, i.e., less non-empty values (assuming that the objects are on average tagged with more than one tag). This approach can be extended to adding those tags to the user profile, which are clearly subsumed by another tag (such as 'RDF' being a sub-topic of 'Semantic Web'). These can automatically be derived, for example, using the GrowBag approach [2] and can further reduce the sparsity of the user profile.
- A lower dimensionality of the user profile if the number of tags is smaller than the number of tagged objects. For this purpose, a controlled dictionary [14] can been derived from the set of all tags. As tags are typically power-law distributed [7], removing the rarely-used tags can reduce the dimensionality of the user profiles by several orders of magnitude (in our experiments, 8600 tags out of 130,000 represented 60% of all occurrences of tags).
- A higher connectivity among the different user profiles as the user profiles are more dense and because the tags in folksonomies tend to be power-law distributed.

In our approach we want to support several different ways of creating user profiles starting from a corpus of tagged objects:

1. Search or navigate through the set of available tags, selecting a subset of the most interesting ones to be able to present the objects associated with this subset of tags, from which the user can select the most interesting ones. This can make use of automatically derived relations between tags as proposed in the GrowBag approach [2].
2. Browsing through the set of objects already existing in the user profile, adding / deleting objects and / or single tags.
3. Browsing through the list of recommended objects (such as publications or persons in the publication domain) and tags and adding the most interesting ones to the profile.

Each user has the possibility to individually modify her profile by adding new objects or removing objects the user is no longer interested in. Also, it should be possible to mark certain topics as 'not interesting': If an object has been tagged by several persons, not all the tags of an object may describe the interests of one particular person. In the publication domain, for example, this means that not all the keywords of a publications with several authors may be relevant for the interests of one particular author; the non-relevant keyword might be referring to a part of the publication written by another co-author.

The tags are typically gained using a manual 'tagging' approach (e.g., in the publication domain, the authors already provide a set of keywords describing their publications). Alternatively, keywords can be retrieved using Information Retrieval methods, for example, from the title, the abstract, or the full text of the publication, though they are typically of lower quality.

3.1 Approaches to Creating and Maintaining User Profiles

Which of the three earlier mentioned ways to creating user profiles are best suited for a particular user strongly depends on the type of user: For users without a profile, we

first try to bootstrap a user profile based on the tags, the user herself has contributed to the folksonomy system (if existing). While this is easy in general folksonomy systems, problems arise in the publication domain because of missing user ids. Hence, it is necessary to match the user name with the names of all authors in the publication dataset and present a list of papers, where the author names match the user name. The user can subsequently process this list to eliminate publications from other authors having the same name.

If a new user has not tagged any objects herself, she can alternatively search the set of available tags to find those tags which best describe her interests. They are used as a conjunctive query to identify a list of potentially interesting publications. To accommodate too large / too small result lists, tags can be added / removed on-the-fly to get a reasonable size of the result list. Tag hierarchies as generated by the GrowBag system [2] can be used to easier navigate through related tags.

After having selected a set of tags, a user can preview and browse the current intermediate user profile comprising the list of objects that are annotated with these tags, adding interesting objects to the user profile or deleting those objects, which are no longer interesting. This also means that the tags associated with this object are added to or removed from the final tag-based profile. This approach enables an automatic assignment of cardinalities in the user profile. For example, if a user has selected five objects as interesting from which three are tagged with 'Semantic Web', the cardinality of the tag 'Semantic Web' in the user profile will be three. In contrast, if the user chooses the interesting tags directly, she would have to assigned the cardinalities manually.

Based on the user profile, the system can also recommend other possibly interesting items or even related tags (cf. Sect. 4). They can be used to further extend and refine the user profile, in case the user agreed with some part or with all recommendations. This is especially useful for people who already work in their community for quite some time and want to monitor the dynamics of the community.

After the user has finished editing her profile we want to export the profile in the RDF format (similar to a FOAF file) which the user can put on her homepage. This allows for an easy exchange of user profiles within a community. Furthermore, other tools can be used to change and maintain the user profile and re-introduce it again to our system later. Hence, we export both the tag-based user profile and also the collection of objects on which the user profile is based. For this purpose, we need unique identifiers for the objects, such as a URL. Moreover, users can also directly view their profile with any RDF viewer and see how their interests overlaps with their colleagues.

3.2 Experimental setup

The TBProfile system applies our ideas to the digital library domain, where the tagged objects are publications and the tags are the keywords, manually annotated by the authors of the publication.

We have used the DBLP collection of around $650,000$ computer science related publications, providing the URLs for about $330,000$ of the publications. As described in [2], all manually annotated keywords were extracted from the provided URLs using a wrapper-based approach. From about 53.000 URLs, proper tags could be found, resulting in a 'folksonomy' of tagged publications with around 130,000 popular unique

tags. All tags were post-processed using acronym replacement (e.g., WWW → World Wide Web) and Porter stemming and the tags which were mentioned less than five times were filtered out. This resulted in a controlled vocabulary of about $8,600$ 'main' tags, representing 60% of all occurring tags due to the power-law distribution of tags.

The TBProfile system comprises also a web application which allows the users to select tags from the controlled vocabulary of tags, either by browsing the set of available tags or by starting from the set of defaultly assigned publications and using the recommender system. For the selected tags, a user can search for publications and select the ones relevant to her current interests. When the user has finished editing her list of publications, she can view her profile and get recommendations about other publications, tags, and persons.

As an example, Table 3 shows the tag-based profile of 'Wolfgang Nejdl', which has been gained only using his publications available in our tagged DBLP collection.

Keyword name	Occurrences	Global Frequency
XML	1	554
UML	1	302
Web services	1	193
Ontology	1	158
Adaptation	1	102
Semantic Web	5	190
Peer-to-peer	4	123
Personalization	4	92
Standards	1	61
Query languages	1	63
Hypermedia	1	93
Generalization	1	25
Web search	1	49
E-learning	1	59
Network management	1	49
Diagnosis	1	49
Ranking	1	31
Pagerank	1	38
Web engineering	1	35
Adaptive hypermedia	2	30
Meta-modeling	1	9
XML scheme	1	23
XMI	1	9
Asynchronous collaboration	1	8
Synchronous collaboration	1	5
Adaptive Web	2	5

Table 3. Tag-based profile of Wolfgang Nejdl

The column 'Occurrences' denotes the number of times the keyword appears in the profile and 'Global Frequency' represents how many times the keyword appears in all publications of the community.

Additionally, we also want to let the users explore different sources for the tags assigned to an object. In the digital library domain, this can be, for example, keywords

derived from the publication title, or keywords derived from the abstracts. While manually created keywords usually have a very high quality, using keywords extracted from the title / the abstract leads to a larger set of tagged documents for the case that not all documents were manually tagged by the authors.

4 Using Tag-Based Profiles for Recommendations

One application of the created user profiles is to provide the user with recommendations about related objects or tags (i.e., to use in regular search engines), and related users with similar interest, who are candidates for collaborations. The main intention is to deeper analyze the research community.

4.1 Basic Idea

The basic idea is to use the tag-based profiles as input to standard recommender system technology [12], to be able to recommend related objects, tags and persons. Hence, we combine the 'user profile' aspect of collaborative filtering systems with the feature-representation aspect of content-based systems. This means, we combine the idea of letting users 'recommend' items, which is a different interpretation of users tagging objects, with the characteristics of legacy information retrieval systems and the derived content-based recommender systems, where objects are represented by their features, typically a vector of terms.

The TBProfile system comprises a user-item recommender system, that computes similarities between users based on a cosine function, that has been extended with the concept of an 'inverse user frequency' [3] as the analogue concept to TFxIDF in the recommender system domain. The similarity between two users $U1$ and $U2$ is computed as shown in Eq. (1)

$$cos\_iuf(U1, U2) = \frac{\sum_i v_{U1}(i) * iuf(i) * v_{U2}(i) * iuf(i)}{\sqrt{\sum_k (v_{U1}(k) * iuf(k))^2 * (v_{U2}(k) * iuf(k))^2}} \quad (1)$$

with $v_U(i)$ being the normalized 'vote' of user U for the item i, and $iuf(k)$ defined as shown in Eq. (2)

$$iuf(k) = log(\frac{\text{number of users}}{\text{number of votes for k}}) \quad (2)$$

As an example, for a user $U1$ having selected three publications for her profile with in total 10 distinct keywords K_{U1}, $v_{U1}(i)$ will be $1/10$ for $i \in K_{U1}$.

The neighborhood N_U for each user U is computed using the k-nearest neighbor approach [13] with $k = 20$. Finally, we compute the recommendation for a certain item I by aggregating the votes of all neighbors of U in a similarity-weighting [6] approach according to Eq. (3)

$$rec(U, I) = \frac{\sum_{j \in N_U} v_j(I) * cos\_iuf(U, j)}{\text{neighborhood size}} \quad (3)$$

The neighborhood size can at most be k, but may be smaller if only very few similar users are found for the given user U.

Our system can provide several kinds of recommendations:

1. Objects based on users.
2. Users based on objects.
3. Users based on co-tagging.
4. Tags based on users.
5. Users based on tags.

In the first case, the recommender system uses a standard user-object matrix to be able to recommend related objects (e.g., publications in the digital library domain [8]). In the second case, the matrix is transposed to be able to recommend users instead of objects. This is one variant to get information about other users in the community. In the third variant, the recommendation is based on a matrix of users having tagged the same objects. This can also be used to get information about people in the community. The fourth case is the first one, where we actually use the tag-based user profiles to create a user-tag matrix and finally recommend tags for the users in that matrix. By transposing this matrix, we are able to recommend users based on the tags users have annotated, which is the last variant described here.

4.2 Experimental setup

Our TBProfile application can give recommendation for publications, keywords and other users of the system. For our experiment we have selected the top 60 authors who have published publications with the topics “semantic web” and “OWL”. For these authors we have built their profiles based on the keywords of the papers they have authored. The intermediate profiles comprised on average 34 publications while the number of keywords per authors was only 16 due to the fact that only 20% of the publications in our database are tagged.

For the profile from Table 3 we show the recommendations in the following tables regarding recommended authors. We only provide the user with at maximum the top ten results.

Table 4 is the result of case 3, i.e., based on a co-author matrix.

Recommended author	score
Rudi Studer	0.0512828
Dieter Fensel	0.0362056
Ian Horrocks	0.0238108
Peter F. Patel-Schneider	0.0221371
Raphael Volz	0.022023
Alexander Maedche	0.0183598
York Sure	0.013157
Timothy W. Finin	0.0268965
Nenad Stojanovic	0.00993426
Enrico Motta	0.00619568
Daniel Oberle	0.0060706

Table 4. Recommendations based on coauthorship

These recommendations clearly focus on the ‘senior’ people, having long lists of publications. In this recommendation, tags have not been used at all. In contrast, the

Recommended collaborators	score
Steffen Staab	0.390822
Axel Polleres	0.311705
York Sure	0.299058
Siegfried Handschuh	0.253242
Nigel Shadbolt	0.214939
Dieter Fensel	0.21334
Ruben Lara	0.206428
Yuan-Fang Li	0.193029
Bijan Parsia	0.187487
Carole Goble	0.17375

(a) . . . based on keywords

Recommended collaborators	score
Siegfried Handschuh	0.411228
Rudi Studer	0.274152
Dieter Fensel	0.137076
York Sure	0.137076

(b) . . . based on publications

Table 5. Recommended collaborators. . .

recommendations based on the tags (cf. Table 5 (a)), are based on the content and are not related to the number of publications. Hence, also 'junior' people are recommended by our main scheme. For comparison, we also show the result of case 2 in Table 5 (b)), where we use the transposed user-publication matrix to recommend users. We can see, that only four persons can be recommended here, for other users of the system this list of recommendations was even empty. This is because the user-publication matrix is in general less connected than the matrix based on the tags as people tend to share tags and use some of them very often (the 'stars' in the power-law distribution).

5 Conclusions and future work

Having a well-defined user profile can be very helpful, especially in research communities where people are explicitly interested in finding out firsthand about what happens in their line of work. No matter if people are interested in finding new relevant publications, related topics or about people to collaborate with, their user profile can support the information flow in their Community of Practice. In this paper, we use the tags from a folksonomy system to build user profiles and feed them to a recommender system, especially to identify related persons in the community. This unique combination of the user profile aspect of collaborative recommender systems with the feature-based schema to describe user profiles (as used in content-based recommender systems) is intended to better capture the interests of the users in the recommendation process and to reduce problems with sparse user profiles. We have shown the TBProfile prototype, implementing a rudimentary system for creating tag-based user profiles in the digital library domain and using a user-item based recommender system to find potential people to extend a user's community of practice. Even though only 20% of the publications in our database are tagged, we have shown evidence that using tag-based profile can give more recommendations than standard object-based user profiles.

For future work, we want to focus mainly on the evaluation of our system, especially involving relevance feedback of real users by notifying them regularly about new interesting publications, persons, and keywords and using answers about the value of

the recommendation to update the user profile. Furthermore, we want to compare the recommendations provided by different tagging schemas (manually tagged vs. automatically derived from the title or the abstract). You can see our current prototype at http://www.l3s.de/∼diederich/TBProfile.

References

1. N. Agarwal, E. Haque, H. Liu, and L. Parsons. Research Paper Recommender Systems: A Subspace Clustering Approach. In *International Conference on Web-Age Information Management (WAIM)*, pages 475–491, 2005.
2. W.-T. Balke, U. Thaden, and J. Diederich. The Semantic GrowBag Demonstrator for Automatically Organizing Topic Facets. In *Proceedings of SIGIR2006 Workshop on Faceted Search*, Seattle, USA, August 2006.
3. J.S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proc. of Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, USA, July 1998. Morgan Kaufmann Publisher.
4. P.-A. Chirita, A. Damian, W. Nejdl, and W. Siberski. Search Strategies for Scientific Collaboration Networks. In *Proceedings of 2nd P2P Information Retrieval Workshop held at the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, Bremen, Germany, 2005.
5. R. Denaux, L. Aroyo, and V. Dimitrova. An approach for ontology-based elicitation of user models to enable personalization on the semantic web. In *Proc. of the International World Wide Web Conference (WWW)*, pages 1170–1171, New York, NY, USA, 2005. ACM Press.
6. J. Herlocker, J.A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4):287–310, 2002.
7. A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, June 2006. Springer.
8. S.M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S.K. Lam, A.M. Rashid, J.A. Konstan, and J. Riedl. On the Recommendation of Citations for Research Papers. In *Proc. of ACM Conference on Computer Supported Cooperative Work (CSCW)*, New Orleans, USA, November 2002. ACM.
9. S.E. Middleton, N.R. Shadbolt, and D.C. De Roure. Ontological User Profiling in Recommender Systems. In *ACM Transactions on Information Systems (TOIS)*, pages 54–88. ACM Press, 2004.
10. M. Montaner, B. López, and J.L. de la Rosa. A Taxonomy of Recommender Agents on the Internet. *Artificial Intelligence Review*, 19:285–330, 2003.
11. D. Pavlov, E. Manavoglu, D.M. Pennock, and C. Lee Giles. Collaborative filtering with maximum entropy. *IEEE Intelligent Systems*, 19(6):40–48, 2004.
12. P. Resnick and H.R. Varian. Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997.
13. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of Recommendation Algorithms for E-Commerce. In *Proc. of ACM Conference on Electronic commerce (EC)*, pages 158–167, Minneapolis, USA, October 2000.
14. F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.