Content- and Graph-based Tag Recommendation: Two Variations

Johannes Mrosek, Stefan Bussmann, Hendrik Albers, Kai Posdziech, Benedikt Hengefeld, Nils Opperman, Stefan Robert and Gerrit Spira

FH Gelsenkirchen, Department of Computer Sciences, Neidenburger Strasse 43, 45877 Gelsenkirchen, Germany,
mrosek@internet-sicherheit.de, stefan.bussmann@gmx.net, joschgg@gmx.de, kaip@gmx.net, hengefeld@web.de, nilsoppermann@web.de, FSMARINE@gmx.de, gerrit.spira@gmx.de

Abstract. We describe two variants of our approach to tackle the task 1 & 2 of the ECML PKDD Discovery Challenge 2009 where each contenter had to identify up to 5 tags for each resource of a given set of either bibtex-like references to publications or bookmarks. The quality of the results was measured against the tags that users of the data source (www.bibsonomy.org) had originally assigned to the resources (F1 measure). In our approach, we either generate tags (from the content of the given resource data or after crawling additional resources) or we request tags from tagging services. We call each of this tag sources a *tag recommender*. We then combine the results of the tag recommenders based on weighting factors. The weighting factors are determined experimentally by comparing generated and expected tags based on the available training data. This general idea is also used for the graph-based approach required to solve task 2. Here again, the final tag recommendations are computed from the individual results of the different tag-recommending algorithms. In the preliminary result list, we ranked second for task 1 (Group 2) and nineth for task 2 (Group 1).

Key words: content-based graph-based tag recommendation bibsonomy

1 Preliminaries

Assigning tags to resources can be an effective instrument to organize an information space. Users interacting with this information space may utilize the tags to identify relevant resources or groups of resources. User-driven tag assignment is a popular way to ensure at least some sort of tag quality[1]. Often, the users assigning tags are supported by algorithmic tag recommendation. This may be as simple as offering auto-complete fields for entering tags that display tags already assigned by users, or it may be based on a full-fledged analysis of the resource

[1] in the sense that the tags indeed help users to find or organize resources, for recent results on tag quality compare [2, 3].

the user wants to tag. This analysis may present a set of automatically identified tags to the user. Within the later context, compare [1], we describe two variants of our approach to content-based tag recommendation which we applied to task 1 and 2 of the ECML PKDD Discovery Challenge 2009. We acted as two teams (with completely independent implementations), each team deploying its own variation of the overall approach – and each with different success. In the following we first describe the solution and results of group 1[2] for tasks 1. This will be followed by a brief presentation of the differences of the solution for task 1 of group 2[3] and their results. We conclude with details of the solution for task 2 of group 1.

2 Content-based Tag Recommendation

This attempt on content-based tag recommendation uses different sources to generate tag candidates. These candidates are combined on the basis of appropriate weighting factors which have been assigned to the particular sources ex ante. The first kind of source are web services, which offer information for known resources. This information contains tags which already have been assigned to those resources. In this case we query del.icio.us and citeulike.org. The second kind of source are the resources themselves. In case of bookmarks the content of the according websites is crawled and analyzed. For bibtex entries the information contained in the bibtex table is taken into account. So tag candidates are determined without using any external service. The last source is also a web service called tagthe.net. It already has an engine, which recommends tags for a given URL.

2.1 Harvesting Tag Candidates

The first step to a tag recommendation is the accumulation of candidates and additional information from the several sources. For every URL in the bookmark table, the del.icio.us service is queried. It can be accessed via a feed. The URL md5 hash of the resource is inserted into a URL-pattern. When the resulting URL is accessed, all available information is returned. It includes a count which specifies how often the URL was posted, and a list of top tags assigned to it. Each tag is supplemented with an information about the frequency with witch it has been assigned to the given resource.
In some cases it is possible to find tag candidates for bibtex entries at citeulike.org. The description or misc field of the bibtex table often contain a citeulike-id. With this id the according citeulike page can be called and crawled for the assigned tags. These tags are already classified as "tag", "publisher" and "author". Numerical information like a count cannot be obtained.
A similar kind of data is provided by the service tagthe.net. It generates classified tags for a submitted URL automatically and independent of the document

[2] Group 1: Mrosek, Bussmann, Albers, Posdziech.
[3] Group 2: Hengefeld, Opperman, Robert, Spira.

format behind it. The available categories are "tag", "author", "person" and "location". Like citeulike.org, tagthe.net returns no numerical information. Anyway this service can be seen as a backup system, because it can provide tag candidates for every URL. Thus it is called for all URLs in the bookmark and the bibtex table. Unfortunately there is no information about the algorithmics of the service available.
In addition to the citeulike-ids the bibtex table contains further interesting information. The title, journal and description fields contain words that can be potentially used as concise tags. Hence after comparison with a stop word list all remaining words in these fields are interpreted as tag candidates.
The last source for tag candidates are the websites behind the URLs in the bookmark table. After crawling and parsing the site's source code the words are counted and checked against a stop word list. Furthermore they are classified by the location of their appearance like in "meta keywords", "meta description", "title" and "body".

2.2 Selection Tags from the Candidates

The recommendation system has a hierarchical structure. This means there is one meta recommender which relies on several separate source recommenders, one for each source described in section 1.1. These recommenders provide up to 20 tag candidates for a queried content id. Every tag candidate is complemented by a score the according recommender assigns to it. This score can vary between 0 and 1. How the recommender constitutes that score depends on the source and is described in section 1.3. After the source recommenders have made their suggestions, the meta recommender takes all of the intermediate results and determines the actual tags. Therefor it combines a candidate's frequency of appearance in all sources k with the score s_i $(1 \leq i \leq k)$ provided by the source recommenders. s_i is already influenced by a weighting factor for the individual sources. How this factor is determined will be described in section 1.4. The final score s for a tag candidate can be determined by the formula

$$s = k \cdot (s_1 + s_2 + ... + s_k). \tag{1}$$

Note that s can be a value greater than 1. When the meta recommender has calculated these aggregated scores for all candidates, they are ordered by this new information. The five tags with the highest scores are selected as recommended tags. In order to prohibit recommendation of tags with a very low score, an optional filter can be set. This helps to enhance the precision.

2.3 Calculating the Individual Scores

As already stated the calculation of the single scores differs depending on the source. The reason for this is the additional information, which is provided besides the tags. Every source offers a different kind of information. To calculate a del.icio.us score $s_{delicious}$ for a tag, the number of posts n which assigned it to the resource is devided by the total number p of posts for the resource at del.icio.us. After that the result is multiplied by the weight $w_{delicious}$ constituted for del.icio.us.

$$s_{delicious} = w_{delicious} \cdot \frac{n}{p} \tag{2}$$

The scores for tagthe.net and citeulike candidates are determined as follows: For each of the provided categories a weighting factor $c_{categorie}$ is defined: $c_{tag} = 1.0$, $c_{author} = 0.3$, $c_{publisher=0.2}$, $c_{location} = 0.2$, $c_{person} = 0.1$. The score $s_{tagthe/cite}$ of each tag candidate is the result of the product of the source's weight $w_{tagthe/cite}$ and the categorie's weight $c_{categorie}$. In one case there is an exception from this practice. As can be detected in the training data the tag "juergen" appears pretty often. So this special tag is always handled with a score of 1.0. Otherwise all scores are determined by

$$s_{tagthe/cite} = w_{tagthe/cite} \cdot c_{categorie}. \tag{3}$$

The tag candidates generated from the bibtex entries are treated in a similar way. For some fields a weighting factor c_{field} is set: $c_{title} = 0.55$, $c_{journal} = 0.25$, $c_{description} = 0.2$. Only tags from the title field are used as a suggestion for the meta recommender. These candidates get a higher score if they also appear in the journal or description field. The individual scores for these fields are added to the initial title-score of 0.55.
The scoring process for the crawled content of a website is a little more complex. It is based on the information about the location of appearance as described in section 1.1. Three steps are passed until the final score is determined. In the first step a tag's frequency of appearance $n_{location}$ in the individual locations is counted. This value is weighted by a factor $c_{location}$ which is related to the location's importance. E.g. a word from the title or the keyword-list is rather a good tag than one from the body. All counts are multiplied by their weighting factors get summed up to a kind of weighted frequency n_{wfreq} for the whole resource

$$\begin{aligned} n_{wfreq} = n_{title} \cdot c_{title} + n_{description} \cdot c_{description} + \\ n_{keywords} \cdot c_{keywords} + n_{body} \cdot c_{body}. \end{aligned} \tag{4}$$

If a tag appears at different locations, its score is raised in step two. This takes into account the fact that such a tag is a good tag in most cases. To raise the score, n_{wfreq} is multiplied by a factor f_{cat} which is related to the number of categories the tag appeared in.

$$f_{cat} = \begin{cases} 1, & 1 \text{ categorie} \\ 1.5, & 2 \text{ categories} \\ 3, & 3 \text{ categories} \\ 5, & 4 \text{ categories} \end{cases} \tag{5}$$

The modified weighted frequency n^*_{wfreq} is determined by

$$n^*_{wfreq} = n_{wfreq} \cdot f_{cat}. \tag{6}$$

In the last step, n^*_{wfreq} is normalized to the common score interval $[0, 1]$. Therefor every n_{wfreq} is divided by the highest n_{wfreq} which is reached for the crawled resource. The result is a score s_{cc} for a tag from the crawled content. The number of tags which are returned to the meta recommender is limited. Only the 20 tags with the highest scores are chosen.

2.4 Calculating the Source Weights

A key point in this two-stage recommendation approach is the calculation of suitable weighting factors for the different sources. Their tag-quality varies in a wide range. Thus handling the candidates of the individual recommenders as if they were of similar quality leads to bad results. As a foundation for the weighting factor calculation the postcore-2 of the training dataset was used. For every source the set of tag assignments it can supply was determined and an according tas file was generated. E.g. to calculate a factor for citeulike, a tas file with all content ids, which are associated with bibtex entries containing a citeulike-id, was generated. Corresponding to the tas files a result file was created for every source. This result was independent of the meta recommender and all the other sources. The result files were measured with f1-score against the tas files created before. The first attempt was to use the f1 score a result file achieved as the weighting factor for the according source recommender.
However when testing the complete recommendation process against postcore-2, experiments with various weighting factors pointed out a better choice. Using the precision value which has been computed for the various result files to get the f1-score, provides better overall results. Thus all the weighting factors for the particular sources assumed in the meta recommender match the precision the respective source recommender reaches for the subset of tag assignments it can supply.

2.5 Results

Table 1 displays the results each recommender achieved for his own subset of the postcore-2 training data. The intermediate results in the first five rows show

recall, precision and f1-score for every source recommender. The weighting factor later used for the meta recommender is the particular precision as described in section 1.4.
Row five and six present the results of the meta recommender for the training data. The first is generated without a filter whereas the second one uses a filter to avoid tag recommendations with very low scores as described in section 1.2.

Table 1. Comparison of the results for test and training data set

Dataset	Recommender	Supplied Posts	Recall	Precision	F1-Score
Training	citeulike.org	5285 / 6372	0.446	0.134	0.206
	del.icio.us	38383 / 40882	0.418	0.353	0.383
	tagthe.net	40468 / 51580	0.066	0.053	0.059
	bibtex	22341 / 22341	0.155	0.127	0.139
	web content	33193 / 40882	0.150	0.123	0.135
	meta	63107 / 64120	0.350	0.254	0.294
	meta with filter	62104 / 64120	0.344	0.269	0.302
Test	**Meta/Overall** with filter	30844 / 43002	0.132	0.103	**0.116**

The table shows that del.icio.us and citeulike produce good tag candidates. Recommendations made by other sources have a much lower quality. The meta recommenders result for the test dataset is far below the result of the training data set.

2.6 Conclusion

The comparison of the meta recommender's results for training and test data leads to the conclusion that the choice of sources was suboptimal. As the training results were computed on the postcore-2 the web services provided tags with a high quality. Every resource was posted in bibsonomy at least two times. So the probability that it is contained in the web services' databases as well, was pretty good. Thus the decrease in score might be explainable by the unpopularity of the resources in the test dataset.
In conclusion the two stage approach is a good attempt for popular resources. To raise the result quality for unpopular resources too, further sources with previously assigned tags have to be added.

2.7 Group 2: Variations for Task 1

Table 2. Comparison of the results for test and training data set

Dataset		Recommender	Recall	Precision	F1-Score
Training	Bookmarks	del.icio.us	0.391	0.280	0.326
		WebCrawler	0.139	0.099	0.116
		DataSet	0.113	0.092	0.102
	Bibtex	WebCrawler	0.250	0.128	0.169
		GoogleScholar	0.087	0.073	0.079
		DataSet	0.083	0.061	0.070
	Meta		0.334	0.213	0.260
Test	**Meta/Overall**		0.214	0.155	**0.180**
	Meta	w CiteULike	0.218	0.157	0.183

Harvesting Tags: In addition to citeulike and del.icio.us, group 2 implemented a tag recommender using Google Scholar for bibtex entries. Using the title data only, links to the resource itself or similar resources were harvested. From the first ten entries three were selected by counting identic words in the titles of the referenced documents (ignoring certain stop words). In the competition, the Google scholar tag recommender harvested roughly 60.000 links for 24.000 bibtex data entries. Subsequently, the Web content crawler of group 2 was used to obtain candidate tags from the three selected resources.

Furthermore, the web content crawler was able (to a certain extend) to parse PDF documents in addition to HTML documents. Also the implementation of the citeulike tag recommender differed: a recent database dump of the citeulike data was used which made it possible to search for DOI-ids or URLs directly and to determine the overall frequency of tags in the citeulike data set. TagTheNet was not used. A straighforward tag recommender (called Data set recommender below) that analysed the relevant fields of the data entries complemented the set of recommenders.

Selecting Tags: No additional filtering for small scores was used. If less than 5 tags were harvested, the remaining slots for tags were filled by randomly drawing tags from the 30 tags most often used in bibsonomy[4]. The weights for a recommender could be different for bibtex and bookmark context. The weights for the

[4] Not a very successful idea – first evaluations show that this was rather counterproductive.

recommenders were chosen manually, based on experiences while experimenting with the training data. Tags recommended by more than one recommender were rated significantly higher by multiplying their score with a factor which depends exponentially on the number of recommendations.[5]

Results: The results of group 2 for task 1 are presented in table 2. Additional quantitative data are shown in table 3.

Please note that due to a persistent problem with the CiteULike tag recommender, it is not listed in the training data and haven't been used in the actual competition. Therefore, the relevant F1-score for the ranking in the competition is **0.180** (second place). Some more details on the impact of the different tag recommenders will be presented at the workshop.

Table 3. Additional quantitative data

	Competition	Training
Tags harvested	1432413	8389379
Bookmarks tagged	16898	263004
Bibtex entries tagged	26104	158924

3 Graph-Based Tag Recommendation

The graph-based approach adapts the same hierarchical recommender structure the content-based approach is based on[6]. This means that there is also one meta recommender which combines the results of subordinated recommenders.
Instead of using external sources, these recommenders implement different algorithms. Every algorithm processes bibsonomy's graph structure and the contained user information in a different way to generate a set of tag recommendations.
Like for the content-based approach every tag is valuated with a suitable weighting factor. This weighting factor also depends on the quality of the subordinated recommender. It is determined in the same way as for the source recommenders in Task 1 (cmp. section 1.4). The used benchmark data set contains the whole postcore-2. For processing the final test data set, three different algorithms were used to supply the meta recommender. All of these algorithms have a different focus on evaluating the graph-structure. The specific operation methods are presented in sections 2.1 to 2.3.

[5] value = value * Math.pow((1.0 + 3*(count - 1)/10), count);

[6] The following section documents the work of group 1 only

3.1 Tag by Resource

The first algorithm selects tags based on the resource information. Therefor all tags which have already been assigned to a queried resource are determined. Additionally each tag's frequency of appearance n_{local} in combination with the resource is counted. To calculate the score s_{tr}, this value is taken and divided by the number n_{post} the resource was posted. The result is multiplied by the weighting factor w_{tr} for this recommender. Like in section 1, the recommenders precision for postcore-2 is used for this purpose.

$$s_{tr} = w_{tr} \cdot \frac{n_{local}}{n_{post}} \tag{7}$$

If two tags reach the same score, the one with the higher frequency of appearance in the entire postcore is preferred.

3.2 Tag by User

The second algorithm recommends and scores tags with a special relation to the user. Algorithmically this approach is very similar to the one described in section 2.1. At first the set of tags is identified which the user who makes a post has assigned to other posts previously. The frequency of usage for the tags n_{local} is also determined. From this set of tags the n_{local}-value for the most used one is taken as a reference value n_{max}. The score s_{tu} this recommender calculates for a tag is as follows

$$s_{tu} = w_{tu} \cdot \frac{n_{local}}{n_{max}} \tag{8}$$

The weighting factor w_{tu} is determined as usual (cmp. Section 2.1).

3.3 Tag by User Similarity

The last algorithm determines tags, which have been used by similar users. The similiarity between two users is defined over the number of equal resources posted by them.
Therefor the first task is the determination of similarities between all users. These are calculated by the Tanimoto-Score under consideration of the posted resources. These resources are used as comparison criterion for different users.
In the recommending process the top five of the similar users are identified. The tags they assigned to the posted resource are accumulated and sorted by their frequency of appearance n_{count} in the posts of the top five users group. As a reference value the maximum frequency of appearance n_{max} of a tag in the same group is utilized. The final score s_{us} for a tag generated by this recommender is calculated with the following formula:

$$s_{us} = w_{us} \cdot \frac{n_{count}}{n_{max}} \tag{9}$$

3.4 Results

The results for Task two are arranged the same way as the results for the content-based approach in task one. The recall, precision and f1-score values for the training datatset are shown individually for every algorithm. Furthermore results are visualized for meta recommenders which use different combinations of the subordinated recommenders. A filter like the one described in 1.2 is tested too.
The recommendations for the test dataset have been made by a meta recommender which uses all of the three subordinated recommenders and a filter. They are a good deal worse than the results for the training dataset.

Table 4. Comparison of the results for test and training data set

Dataset	Recommender	Supplied Posts	Recall	Precision	F1-Score
Training	1. by Resource	64120	0.731	0.586	0.651
	2. by User	64120	0.349	0.205	0.258
	3. by User-Sim.	34738	0.265	0.271	0.268
	1. + 2. + 3	64120	0.774	0.517	0.620
	1. + 2.	64120	0.846	0.570	0.681
	1. + 2. with filter	64120	0.846	0.576	0.685
Test	1. + 2. + 3. with filter	778 / 778	0.389	0.262	0.313

3.5 Conclusion

Like for task 1 there is a dramatic decrease of the f1-score in the test. In this case the explanation can be found in the composition of the training data files. When evaluating the methods for task 2 with the training data, the post, a recommendation is made for, is not excluded from the dataset. Thus the tags of the post are definitely in the training dataset. This increases the probability that the expected tags are recommended. As a consequence, the scores are higher as for the test data which are not included in post-core 2.
In conclusion, the evaluation of the training data is not valid, but the algorithm works correct for the test data.

4 Final remarks

The competition offered an excellent test bed for our approach to tag recommendation. Though we are pretty happy with the results obtained, much work remains to be done:

- setting the weights for the different recommenders could be improved (iterative algorithmic optimisation; cleaning/choosing training data),
- relations to similar resources in the bibsonomy data set (identified for example with the help of the recommended tags or based on Web crawling / Google scholar results) could be explored: if, for a given resource x, a *similar* resource y in the bibsonomy data sets is identified, tag candidates can be drawn from resource y (either directly or via further graph-based or recursive similarity-based analysis),
- fine-tuning of the individual tag recommenders, for example by coupling the Web crawler to the tag database,

to name just a few open topics. We want to thank the organizers[7] and to express our hope that this stimulating and exciting competition will be continued in the future and that some of our results and suggestions may help others to develop further improved tag recommenders.

References

[1] Jäschke, R., Marinho, L.B., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag Recommendations in Folksonomies. In: PKDD 2007, Springer (2007)

[2] Sen, S., Vig, J., Rield, J.: Learning to Recognize Valuable Tags. In: IUT'09, ACM (2009)

[3] Zhang, S., Farooq, U., Carroll, J.M.: Enhancing Information Scent: Identifying and Recommending Quality Tags. In: GROUP'09, ACM (2009)

[7] and the anonymous reviewers and Wolfram Conen for their valuable comments.