Labeling Clusters - Tagging Resources

Korinna Bade and Andreas Nürnberger
Otto-von-Guericke-University Magdeburg
D-39106 Magdeburg, Germany
{korinna.bade,andreas.nuernberger}@ovgu.de

Abstract

In order to support the navigation in huge document collections efficiently, tagged hierarchical structures can be used. Often, multiple tags are used to describe resources. For users, it is important to correctly interpret such tag combinations. In this paper, we propose the usage of tag groups for addressing this issue and an algorithm that is able to extract these automatically for text documents. The approach is based on the diversity of content in a document collection and a user's structuring preference. For evaluation, we use methods from ontology evaluation and show the validity of our approach on a benchmark dataset.

1 Labeling vs. Tagging

When searching for information, structured access to data, e.g., as given by web directories or social tagging systems like del.icio.us[1] can be very helpful. The goal of our work is to automatically provide such structure for unstructured collections. Here, we consider text documents. One possible direction is to hierarchically cluster documents based on their content and further knowledge like personal clustering preferences [Bade and Nürnberger, 2008]. This hierarchy can then be used to browse the collection. However, the labeling of the individual clusters is crucial. A standard approach is to select a small set of terms that are expected to describe the documents in the cluster well.

The idea of cluster labeling is closely related to tagging. In tagging, a small set of terms is assigned to a resource with the goal to describe it. Theoretically, every resource can be described with a different set of terms. However, existing tagging systems have shown that users tend to use similar / equal terms [Golder and Huberman, 2006]. This makes finding relevant or interesting resources more easy for a user. Therefore, a system for automatic tagging should follow this idea. An initial clustering identifies similar documents having common properties. It is natural to tag the documents of a cluster with similar tags. Hence, the cluster label can be used to tag all resources in this cluster. Furthermore, hierarchical relations between the tags can be derived from a cluster hierarchy. Automatic tagging was already proposed in the literature, e.g. by [Mishne, 2006; Begelman *et al.*, 2006].

Summing up, access to the initially unstructured collection can be given either through a labeled cluster hierarchy or through a tag cloud as typical for todays tagging systems. Both representations can be gained through the same algorithm. One possible solution is presented in the following, which is based on an initial hierarchical clustering. This paper focuses on the extraction of the tags assuming the cluster hierarchy was already build. Information on the clustering process can be found in [Bade *et al.*, 2007].

[1] http://del.icio.us

2 How to Tag

In todays tagging systems, resources are tagged with one or more single words to describe them. Between tags, usually no relations are assumed (an exception are hierarchical relations from bundle tags). However, multiple tags might be used for two different reasons. First, a user wants to provide synonyms such that more people find his resource. Second, he wants to show that this resource actually belongs to an overlap of several topics. While browsing with a single tag is sufficient for finding a resource of the first type, the second case requires combining more tags.

For cluster labels, it is even more important to know how different tags shall be interpreted. Two terms of a cluster label could correspond either to the same document or to different documents. This implies that documents in the cluster could either belong to the intersection of two topics or that some documents in the cluster belong more to one topic and the others more to the other topic. As an example consider the example in Figure 1, where a cluster is tagged with *banking* and *programming*. This can either mean that the cluster contains documents about banking software or that the cluster contains documents that deal with banking and others that deal with programming. Clusters on a deep hierarchy level are usually very specific and, therefore, can easily be described by a single tag. Several tags are usually used to provide clarification of one concept. For more general clusters, a single tag might not be sufficient to express the scope of the cluster. Therefore, multiple tags can include both, several terms of one concept and terms describing different concepts.

To help in the interpretation of tags, our approach tries to group tags based on their relevance for documents. A group of tags implies that all tags therein describe a document of this cluster together or even better, one dominant concept in the cluster. Such a group, therefore, contains synonyms as well as combined topics. Furthermore, tags in different groups are supposed to relate to different documents in the cluster and with this to different concepts. In the following, we write such a cluster label as a set of tag groups, where each tag group is a set of tags. For the example above, we would have either a cluster label with a single tag group $\{\{banking, programming\}\}$ or a cluster label consisting of two tag groups $\{\{banking\}, \{programming\}\}$. In the next section, we present an approach that is able to determine

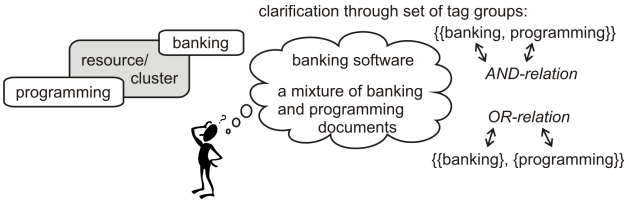

Figure 1: Tag group example

such tag groups given a hierarchical cluster structure.

3 Automatic Hierarchical Tagging

The labeling of a cluster hierarchy is accomplished in three steps, i.e. candidate ranking, grouping, and refinement.

Candidate Ranking First, candidate terms are extracted for each cluster in the hierarchy from the resources assigned to this cluster. Then, these terms are weighted based on their descriptiveness (i.e. their value in describing the cluster) to identify the best tags. A good label should not only describe the resources of a cluster in a condensed form but also distinguish a cluster from others. For efficient browsing in a hierarchy, a label must be able to distinguish a cluster from its sibling clusters as well as show the differences between the cluster and its parent and child clusters. In the literature, some ideas on how to model these properties into a score can be founded, e.g. in [Glover *et al.*, 2002a; 2002b; Treeratpituk and Callan, 2006]. [Geraci *et al.*, 2006] proposed a modified information gain. As we also used this measure, we give its definition here:

$$IG_{mod}(t,n) = \frac{df_{t,n}}{|n_p|} \cdot \log_2 \frac{df_{t,n} \cdot |n_p|}{df_{t,n_p} \cdot |n|} + \frac{df_{\bar{t},n}}{|n_p|} \cdot \log_2 \frac{df_{\bar{t},n} \cdot |n_p|}{df_{\bar{t},n_p} \cdot |n|} \quad (1)$$

Furthermore, we integrated similar ideas in the descriptive score DS_w, originally proposed in [Bade *et al.*, 2007]. Here, we compare this score with the modified information gain and a ranking purely by document frequency df as a simple baseline. In specific, the descriptiveness DS_w of a term t in node n is computed by

$$DS_w(t,n) = \log_2 \left(\frac{rank_{df}(t,n_p)}{rank_{df}(t,n)} \right) \cdot \frac{1 - SI(t,n) + SI(t,n_p)}{2} \cdot \left(\frac{df_{t,n}}{|n|} \right)^w \quad (2)$$

$$SI(t,n) = \frac{\sum_{n_c \in ch(n)} \frac{df_{t,n_c}}{df_{t,n}} \log_2 \frac{df_{t,n_c} \cdot |n|}{df_{t,n} \cdot |n_c|}}{\log_2 \frac{|n|}{min_{n_c \in ch(n)} |n_c|}} \quad (3)$$

with $rank_{df}(t,n)$ being the rank of t in n if terms are ordered by their document frequency in n, $df_{t,n}$ the document frequency of t in n, n_p the parent node of n, and $ch(n)$ the set of child nodes of n. This score combines three factors: The first measures the boost of document frequency ranking in comparison to the parent. This assures that terms get higher scores if they were not already good descriptors for the parent and are therefore too general for the current cluster (as proposed by [Treeratpituk and Callan, 2006]). The second factor considers information on how the term is distributed in sibling and child nodes. SI is based on the KL-Divergence between the distribution of document frequency and the distribution of node size, normalized to stay in the interval $[0;1]$. This means that SI becomes zero, if t is distributed in the child nodes with the same distribution as the documents, i.e. if $df_{t,n_c}/df_{t,n} = |n_c|/|n|$ for all child nodes. On the other hand, SI reaches the maximum of 1, if t occurs only in the smallest child node. In the case of no child nodes (i.e. n is a leaf), SI is also set to 1. By this, the second factor favors terms that occur in several child clusters and therefore generally describes them and penalizes terms that could be also descriptors in sibling nodes. The last factor considers the document frequency as a relatively high frequency is necessary however not sufficient for a good term. How strong the influence of the frequency should be on the final score is controlled by w. Our experiments showed that 0.33 is a good value for w (at least for the considered dataset).

Grouping In the grouping step, the ranked term list is handled sequentially to create tag groups. The first term forms the first tag group. For every following term, it is decided whether it forms a new tag group or belongs to an existing one. A tag group j is hereby represented as a coverage vector cv_j over the documents d_i in the collection:

$$cv_j = \begin{pmatrix} cv_j(d_1) \\ \vdots \\ cv_j(d_n) \end{pmatrix} \quad (4)$$

A document is covered by a term, if the term occurs in it. For the coverage of a tag group, we distinguish between two cases. The binary coverage weight solely distinguishes whether a document is covered or not:

$$cvb_j(d_i) = \begin{cases} 1 & \text{if } g \text{ covers } d_i \\ 0 & \text{else} \end{cases} \quad (5)$$

Different decision boundaries (like the match of a single term) can be used. The weighted coverage of a tag group is a summation of the individual term coverages, whereby the impact of each term is weighted according to its rank in the tag group with an exponentially decreasing influence:

$$cvw_j(d_i) = \sum_{t \in g \cap d_i} e^{-0.5 \cdot (\text{rank}_g(t)-1)} \quad (6)$$

This should ensure that words like stop words that occur in many documents but do not carry much meaning do not join all tag groups. Similarity between a term and a tag group (or two tag groups) is computed by the Dice coefficient between the weighted coverage vectors to measure their overlap:

$$\text{sim}(cv_1, cv_2) = \frac{2 \cdot cv_1 \cdot cv_2}{||cv_1|| + ||cv_2||} \quad (7)$$

A term is merged to the tag group with highest similarity, if this similarity is above a threshold. Once all terms have been assigned to a tag group, the algorithm continuously iterates over the set of tag groups. Two tag groups are merged, if their similarity is still above the threshold. Here, values about 0.6 showed a good performance. This is continued until all tag groups are more dissimilar than the threshold. In the remaining tag groups, it might occur that some tag groups are actually subsets of other tag groups. These are removed in final step. A tag group 1 includes a tag group 2, if the following function of their weighted coverage vectors is close to one:

$$\text{incl}(cv_1, cv_2) = \frac{\sum_i \min(cv_1(d_i), cv_2(d_i))}{\sum_i cv_2(d_i)} \quad (8)$$

Hierarchical Refinement In a final step, specific tag groups from deeper hierarchy levels are propagated up in the hierarchy, because a label extracted for a cluster on a higher level might fail to cover a smaller sub-group of resources. If the extracted label for a non-leaf cluster has a coverage below a certain threshold (e.g. lower than 0.9), refinement is attempted. This is determined through a binary coverage vector that combines the binary coverage vectors of the individual tag groups. Binary vectors are necessary because weighted vectors from different hierarchy levels are not directly comparable. Weighted vectors are transformed to binary ones by setting all dimension to 1 which have larger values than about 0.15. As long as the coverage is to low, tag groups from direct child nodes are added to the parent label. Tag groups with the highest increase in coverage are added first to keep the number of added tag groups small.

4 Evaluation

As the goal of labels or tags is usually to give a good description for a human, automated evaluation is difficult. In particular, it is often the case that there is not one single best solution as language is ambiguous. However, user assessed quality is easily prone to subjectivity which can bias the results. Therefore, we try to gain objectivity in the evaluation by using a benchmark dataset, which represents the ideal solution. This approach is also taken in work dealing with cluster labeling [Glover *et al.*, 2002b; Treeratpituk and Callan, 2006]. However, there are no established standard measures for this kind of evaluation yet. Due to the problems mentioned before, this should give at least a lower bound on the algorithm's performance. This bound can be improved by integrating knowledge about the language. For example, [Treeratpituk and Callan, 2006] used Wordnet to integrate synonyms. Furthermore, one can also imagine the use of additional linguistic relations like hyponyms and hyperonyms to extend the ground truth and thus make the evaluation more accurate. The measures in this work do not make use of linguistic resources. Nevertheless, linguistic resources could easily be integrated, which, however, is left to future work.

In contrast to the work mentioned above, we used evaluation measures from ontology learning [Dellschaft and Staab, 2006], which we modified to fit the purpose of label evaluation. This was necessary because earlier work did not consider tag groups. Furthermore, the measures proposed below are also applicable to standard cluster labeling tasks and therefore also represent an alternative for label evaluation. The basis is built by the computation of the f-score between average precision and recall. Precision π and recall ρ are computed cluster specific, comparing the learned set of tag groups G_l with the reference set of tag groups G_r:

$$\pi(G_l, G_r) = |G_l|^{-1} \sum_{g_l \in G_l} \max_{g_r \in G_r} \mathrm{sim}(g_l, g_r) \qquad (9)$$

$$\rho(G_l, G_r) = |G_r|^{-1} \sum_{g_r \in G_r} \max_{g_l \in G_l} \mathrm{sim}(g_l, g_r) \qquad (10)$$

For each tag group in one set, the best match in the other set is determined and evaluated. The computation integrates a similarity sim of individual tag groups. By using different similarity measures, we evaluated the approaches on different levels. In specific, we used a term based (tb), a rank based (rb), and a document based (db) similarity. The underlying idea of these similarity measures is shown in Figure 2. As can be seen, the term based measure sim_{tb} looks for a perfect match in the terms that would be displayed to a user. In our case, these were the five best terms of a tag group. It is computed by:

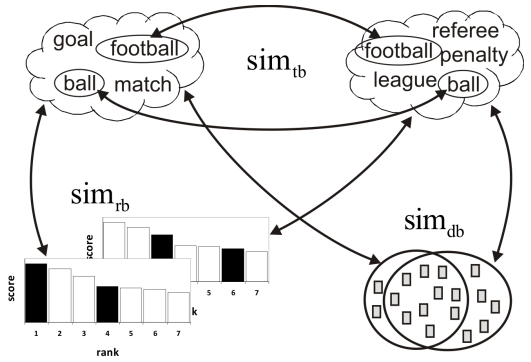

Figure 2: Tag group similarities

$$\mathrm{sim}_{tb}(g_l, g_r) = \frac{|g_r \cap g_l|}{|g_r|} \qquad (11)$$

By definition, this measure is quite strict as it mainly distinguishes between correct and incorrect tags. Although it measures the benefit for the user, it is not very distinctive for comparing different methods. The rank based measure sim_{rb} therefore rather compares the ranking of individuals terms inside the tag groups:

$$\mathrm{sim}_{rb}(g_l, g_r) = \frac{\sum_{t \in g_r \cap g_l} \frac{1}{\mathrm{rank}(t, g_l)}}{1/1 + \cdots + 1/|g_r|} \qquad (12)$$

For this measure all terms are considered which allows a more detailed comparison, showing how "close" the approach came to the reference result. Please note that no ranking is distinguished in the reference tag group.

The document based measure sim_{db} evaluates on an even higher abstraction level. It determines whether two tag groups describe the same set of documents while ignoring the actual terms. This can be seen as a first step in finding the correct tags as the terms are selected based on the documents covered by a tag group. This measure is computed with the Dice coefficient, in this case between binary coverage vectors of the tag groups:

$$\mathrm{sim}_{db}(g_l, g_r) = \frac{2 \cdot cv_{g_l} \cdot cv_{g_r}}{||cv_{g_l}|| + ||cv_{g_r}||} \qquad (13)$$

We evaluated our approach with the banksearch dataset [Sinka and Corne, 2002]. It consists of 11000 web pages in a two layer hierarchy of 14 classes, 4 on the higher level, 10 on the lower level. We extracted three different hierarchies from this dataset: First, the original hierarchy was used with each class having a label consisting of a single tag group (consisting of one or two terms). Second, we build a binary version of the original hierarchy. This resulted in the insertion of intermediate nodes that were labeled with multiple tag groups according to the combined classes. And third, we created a noisy hierarchy in which groups of documents are moved to other classes. The noisy classes were also labeled with multiple tag groups according to the combined instances.

Based on the three hierarchies, we compared two different approaches. First, we applied the standard approach of

Table 1: Results with three f-score measures on three datasets

APPROACH	RANKING MEASURE	ORIGINAL			NOISE			BINARY		
		tb	rb	db	tb	rb	db	tb	rb	db
Single tag group	df	0.5000	0.5377	0.9794	0.4905	0.4751	0.8923	0.3745	0.4593	0.9464
	IG_{mod}	0.8214	0.6920	0.9467	0.7757	0.5992	0.8672	0.7249	0.7404	0.9204
	$DS_{0.33}$	0.7857	0.8003	0.9530	0.7035	0.6458	0.8684	0.6832	0.7556	0.9226
Multiple tag groups	IG_{mod}	0.7775	0.6561	0.9316	0.7959	0.5953	0.8873	0.7233	0.7384	0.9099
	$DS_{0.33}$	0.7932	0.7962	0.9340	0.7762	0.7240	0.9043	0.6658	0.7791	0.8912

using a single tag group formed through a term ranking. This was compared against our approach described in the previous section that tries to build multiple tag groups. The initial term ranking was computed through three different measures, i.e., document frequency df, modified information gain IG_{mod} and our descriptive score DS.

Our results are summarized in Table 1. The three ranking measures are all capable to group the right documents together, as can be seen by the document based measure. However, using document frequency as a measure fails to rank the good terms high, as can be seen by the large drop in performance for the term and rank based measure. Modified information gain and our descriptive score both work quite well. The modified information gain is slightly better considering only the first 5 terms (i.e. in the term based measure). This is especially true for the approach with a single tag group. For multiple tag groups, the difference is much less. Nevertheless, the rank based measure shows that our descriptive score usually ranks the important terms higher than the modified information gain.

Comparing both approaches, it can be seen that performance drops when the approach is restricted to a single tag group but the clusters naturally consist of more than one tag group. Our grouping method can increase the performance for these datasets, especially in combination with our descriptive score. As the original hierarchy only contains single tag groups, it can be used to evaluate whether the algorithms extract to many tag groups. While performance drops for the modified information gain, the performance with the descriptive score is stable in this setting, indicating a good grouping behavior.

5 Conclusion

Concluding the paper, we want to point out that we propose in this paper to improve the effectiveness of tagging in general as well as cluster labeling by integrating relations between tags in form of tag groups. Furthermore, we developed a method that is capable of extracting such tag groups automatically. The presented method is independent of the measure used to find candidate terms, although different measures do not behave equally well as shown in our evaluation. Additionally, we propose some measures to evaluate cluster labeling with a benchmark dataset, for which no standard measure exists yet. In future work, we aim at improving the descriptive score used for initial term ranking. The current score does not yet reflect the idea of multiple tag groups. On the contrary, it assumes that always a single tag group can be found. Therefore, we believe the current results can be improved considering this aspect. Furthermore, we want to analyze the impact of the different threshold parameters in the proposed tagging algorithm. It is our hope that these parameters can be defined collection independent.

References

[Bade and Nürnberger, 2008] K. Bade and A. Nürnberger. Creating a cluster hierarchy under constraints of a partially known hierarchy. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 13–24, 2008.

[Bade *et al.*, 2007] K. Bade, M. Hermkes, and A. Nürnberger. User oriented hierarchical information organization and retrieval. In *Machine Learning: ECML 2007*, pages 518–526, 2007.

[Begelman *et al.*, 2006] Grigory Begelman, Philipp Keller, and Frank Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Proceedings of the WWW 2006 Workshop on Collaborative Web Tagging*, pages 22–26, 2006.

[Dellschaft and Staab, 2006] Klaas Dellschaft and Steffen Staab. On how to perform a gold standard based evaluation of ontology learning. In *Proc. of 5th Int. Semantic Web Conference*, pages 228–241, 2006.

[Geraci *et al.*, 2006] F. Geraci, M. Pellegrini, M. Margini, and F. Sebastiani. Cluster generation and cluster labeling for web snippets. In *Proc. of the 13th Symposium on String Processing and Information Retrieval*, pages 25–36, 2006.

[Glover *et al.*, 2002a] E. Glover, D. Pennock, and S. Lawrence. Inferring hierarchical descriptions. In *Proc. of 11th Int. Conference on Information and Knowledge Management*, pages 507–514, 2002.

[Glover *et al.*, 2002b] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pennock, and G. W. Flake. Using web structure for classifying and describing web pages. In *Proc. of the 11th Int. Conf. on World Wide Web*, pages 562–569, 2002.

[Golder and Huberman, 2006] Scott A. Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[Mishne, 2006] 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, 2006.

[Sinka and Corne, 2002] M. Sinka and D. Corne. A large benchmark dataset for web document clustering. In *Soft Computing Systems: Design, Management and Applications, Vol. 87 of Frontiers in Artificial Intelligence and Applications*, pages 881–890, 2002.

[Treeratpituk and Callan, 2006] P. Treeratpituk and J. Callan. Automatically labeling hierarchical clusters. In *Proc. of the International Conference on Digital Government Research*, pages 167–176, 2006.