Topic-Based User Models: Design & Comparison

Jörg Diederich Wolf-Tilo Balke
L3S Research Center, Leibniz Universität Hannover, Germany
{diederich, balke}@l3s.de

Abstract

User models for browsing digital libraries have to reflect each individual user's interests in sufficient detail while still being of manageable dimensionality for efficiently comparing different users. Using a collection of computer science publications we explore ways to derive topic-based profiles and show how to compare them.

1 Introduction

Today ontologies and classification systems are the driving force behind organizing large document collections and digital libraries. Moreover, if exploited for personalization tasks like alerting or navigational searches, they often enable a better usability of large collections than mere keyword searches. This is because their hierarchical structure is not only useful to organize collections, but can also characterize users by means of *user profiles*. Such profiles describe a user's individual interests in terms of preferred topics together with a numeric value stating to what degree a user is interested in this topic, so called *histogram vectors* [5].

Profiles for specific users can be implicitly created, for example, from users querying and accessing documents annotated with keywords. For this purpose, we plan to use the logging facilities in *FacetedDBLP* (`http://dblp.l3s.de`), our novel faceted browser for the DBLP collection (offering a large range of computer science related literature) first demonstrated at JCDL 2007 [2] and now also offered as a search interface within the DBLP portal (`http://dblp.uni-trier.de/db`).

For using such user profiles (e.g., to compare two profiles or to create the aggregate profile of a sub-community), however, it is necessary to agree on a sufficiently small set of topics (the *topic space*) for the community of users to achieve a sufficient overlap among profiles[1]. After describing related work in Sect. 2, we present the two main contributions of this paper:

- We show two use cases to reduce the topic space dimensionality of topic-based user profiles and present some preliminary experimental data (Section 3).
- We present a new scheme to compare topic-based user profiles, which also considers the depth of the underlying topic hierarchy (Section 4).

Both underline the usefulness of topic hierarchies (even automatically created with a potentially lower quality) in two important aspects of user modeling (design of topic-based user profiles & comparison of user profile instances).

2 Topic Hierarchies in Digital Libraries

Although the benefits of good classifications sytems for digital libraries (like the Dewey Decimal Classification (DDC) system, `http://www.oclc.org/dewey`) are obvious, building and maintaining suitable topic ontologies, however, is still an expensive and mostly manual process. But for many applications such full-fledged and expensive ontologies are not really needed. Especially when it comes to searching and distinguishing between document topics in a structured way, simpler topic hierarchies have proven to be already sufficient [11].

Still, a major challenge is to *automatically* derive such topic hierarchies tailored for specific domains or communities. We advocate using annotations from large corpora of documents related to the domain, such as DBLP for computer science or the Medline database (`http:`

[1] In this respect, topics are a subset of the most popular keywords associated with objects in digital libraries.

//medline.cos.com) for the area of medicine. Often these corpora do not need expensive full-text processing, but already come with metadata annotations. In fact, today tagging documents and creating 'folksonomies', e.g., from author keywords in research papers (Connotea, http://www.connotea.org), topics of Web pages (del.icio.us, http://del.icio.us), or annotations of non-textual documents like images (flickr, http://www.flickr.com), has already become commonplace.

For structuring topic information on Faceted-DBLP, we use our Semantic GrowBag algorithm [1]. It grows domain-specific topic hierarchies from any set of keywords from suitable document corpora. The first step in creating a facet for some topic is to identify all relevant related topics to a query keyword and encourage or discourage the creation of simple subsumption relations by higher order co-occurrences of keywords. Investigating co-occurrences of all keywords in the corpus' documents to find an initial set of related topics, we then use a biased PageRank to efficiently identify the most important topics and their relations for a given community within a given time span. Hence, even current trends in certain topic areas can be reflected. Cross-checking the PageRank scores of related topics, GrowBag also provides confidence scores for all identified relations. The details of the algorithm are beyond the scope of this paper and are presented in [1].

In this paper we will use our DBLP++ collection to illustrate our concepts. The DBLP++ database is built from the DBLP metadata in XML notation (http://dblp.uni-trier.de/xml), enhanced with abstracts and authors keywords as available publicly on the Web. It comprises a total of about 886.000 publication records with their authors, the conference or journal, URLs to electronic versions, abstracts and author keywords (if available). Our dataset comprises roughly 340.000 publications with abstracts, from which about 100.000 are annotated with authors keywords and 68.000 are annotated with ACM classifiers.

3 Deriving Topic-based User Models

In this section, we show-case two methods to find the interesting topics to be used in topic-based user profiles in order to limit the size of the topic space. These methods are based on

1. a predefined classification system the community has agreed upon,
2. automatically created topic hierarchies derived from author keywords, as created by our GrowBag approach.

3.1 Using Pre-Existing Classification Systems

As a first use case we relied on the ACM Computing Classification System (CCS, http://www.acm.org/class/1998) and therefore considered only ACM publications to represent the interests of a user with respect to this classification. Table 1 shows an extract of the ACM CCS, which consists of at maximum four levels of hierarchy (e.g., B. *topic*, B.1 *topic*, B.1.1 *topic*, and B.1.1 *topic* Subject: *subject*).

Table 1: Excerpt of the ACM CCS

A. General Literature
A.0 General
Subjects: Biographies / autobiographies
Subjects: Conference proceedings
Subjects: General literary works (e.g., fiction, plays)
...
B Hardware
B.0 GENERAL
B.1 CONTROL STRUCTURES AND MICROPROGRAMMING
B.1.0 General
B.1.1 Control Design Styles
Subjects: Hardwired control
Subjects: Microprogrammed logic arrays
Subjects: Writable control store
...

Because the ACM CCS has 1420 different topics (including all levels), each user can be characterized by a normalized (typically sparse) 1420-dimensional histogram vector (cf. similar approaches for routing in peer-to-peer networks [9] or for user profiles based on ODP using binary vectors [7]). Hence, each vector element represents the percentage of papers selected by a user classified into this topic. Table 2 illustrates a simplified example histogram vector derived from 40 ACM papers that are classified into 2 topics per paper. For simplicity, only the top-level ACM topics are used to characterize a user in this example.

3.2 Using GrowBag Graphs

The main problem when using author keywords is to derive a controlled vocabulary (the topic space) from the available keywords that is small enough to create reasonably-sized schemas for

Table 2: Simplified Example Histogram Vector

ACM topic	Number of publications	Vector value
A. General Literature	1	0.0125
B. Hardware	0	0
C. Computer Systems Organization	3	0.0375
D. Software	4	0.05
E. Data	2	0.025
F. Theory of Computation	13	0.1625
G. Mathematics of Computing	2	0.025
H. Information Systems	38	0.475
I. Computing Methodologies	10	0.125
J. Computer Applications	2	0.025
K. Computing Milieux	5	0.0625
Sum	80	1

user profiles (in contrast to, for example, [3]). In our DBLP++ dataset, for example, there are 103,000 documents annotated with about 532,000 keywords, of which 194,000 keywords are unique. This is way too large to create profiles, hence we took the following initial approach in our system to reduce the number of keywords:

1. Replace all occurrences of acronyms with 2-5 letters by their full text version. For this purpose, we used a simple regular expression matcher looking for keywords like 'World Wide Web (WWW)' and extracted the full text variant that occurs most often (which also automatically removed misspellings in most cases).

2. Use Porter stemming [10] on the keywords.

3. Completely remove all keywords that occur less than five times to cut the tail of the power-law distribution of keywords.

The acronym replacement in step 1 decreased the overall number of keywords by only about 300 for several reasons:

- There were only about 6,600 potential acronyms in the corpus, of which 2,800 occur more than once and only 1,800 more than twice.

- Some acronyms are never used as full-text in keywords (e.g., MPEG or XML)

Step 2 (Porter stemming) reduced the set of unique keywords from 194,000 to 176,000 (−10%) and the final step reduced it further down to 14,000. However, this is still very large and contains quite some keywords that are not really relevant due to being very specific, and thus might not be useful to characterize users.

Hence, we propose to use our GrowBag graphs, i.e., automatically created hierarchies of keywords, to find the most relevant keywords for a community. We considered and evaluated two possibilities:

1. Filter all those keywords, which are subsumed by other keywords

2. Keep only those keywords which are subsuming other keywords and which are not subsumed themselves.

As our GrowBag graphs are computed for a specific period of time, (i.e., taking only publications within that period into account) Table 3 shows the results for a set of sample periods for both options (for option 1, we present the decrease in the numbers of keywords while in option 2 we show the remaining keywords after filtering).

Option 1 clearly does not help much as it only reduces the topic space by about 5% for the five-year periods and 3% for the two year periods. The main problem is that in this option only few keywords are at all involved in any subsumption relations (cf. column 3). Thus, all those keywords that are in no hierarchical relation, are still left in the topic space. Option 2 reduces the topic space to a more suitable size as it only considers keywords being at the top of a (non-empty) hierarchy. Furthermore, it can also be seen that too short periods might lead to too many keywords filtered since we cannot compute many GrowBag graphs because of too few annotated documents being available in the corpus.

4 Comparing User Profiles

Comparing user profiles is valuable for a range of applications, for example, to find users with similar interests or to match the profile of individual users to a community profile to find out about community membership.

In this section, we present a method to compute the similarity between two user profiles using a quadratic form distance (QFD) measure for histograms, as e.g., used in multimedia databases [4]:

$$D_{ij} = (X_i - X_j)^t * A * (X_i - X_j) \qquad (1)$$

with X_i, X_j being the user profiles for users i, j, respectively, A being the similarity matrix (also called crosstalk matrix) and D_{ij} representing the distance between the profiles X_i and X_j.

Table 3: Reduction of keywords using GrowBag graphs

Year range	Total keywords	Keywords involved in subsumption	Option 1	Option 2
2002-2006	13324	2359	$-810 \sim -6\%$	$1240 \sim -81\%$
1997-2001	11754	1784	$-593 \sim -5\%$	$963 \sim -82\%$
2005-2006	11693	987	$-401 \sim -3\%$	$515 \sim -96\%$
2004-2005	11617	921	$-362 \sim -3\%$	$463 \sim -96\%$
2003-2004	11096	824	$-325 \sim -3\%$	$415 \sim -96\%$
2002-2003	10430	715	$-301 \sim -3\%$	$350 \sim -97\%$
2001-2002	9441	558	$-220 \sim -2\%$	$274 \sim -97\%$

The similarity matrix $A = (a_{ij})$, with i, j being two topics of the controlled vocabulary (be it a pre-specified classification system or a GrowBag graph), has to be defined once from the controlled vocabulary. The main idea is to make the similarity between two topics depending not only on the closeness within the hierarchy, but also on the distance to the root. The main assumption here is that the larger the distance to the root of the tree, the more specific is a topic and the better it characterizes a user.

As an example, we use the ACM CCS such that $A = (a_{ij})$ becomes a $1420x1420$-sized matrix with i, j iterating over all 1420 ACM topics. We have created the similarity matrix according to the following simple rules:

- If i and j are the same, the similarity is the level of i multiplied with $1/(\text{max\_tree\_depth}) = 0.25$. For example, for two topics of the form B.1.2 *topic* Subject: *subject* (level 4), the similarity becomes 1 whereas two topics of the form B.1 *topic* (level 2) achieve a similarity of 0.5 only. Such a relation of the similarity to the level of the topic within the classification system is important: Otherwise, publications which have not been classified very accurately (e.g., using a level-2 topic like B.1 only), become very similar to all other publications, which were also not accurately classified.

- If i and j have a common predecessor topic p in the topic hierarchy, the similarity is the level of p multiplied by $1/(\text{max\_tree\_depth}) = 0.25$. For example, for the topics B.1.3 *topic* (level 3) and B.1.2 *topic* (level 3), the similarity is 0.5 since their common predecessor is $A.1$ (level 2). As another example, the topics B.1.2 *topic* (level 3) and B.1 *topic* (level 2) also achieve a similarity of 0.5 since the latter topic constitutes the common predecessor.

However, there is a special case, which can also be seen in table 1: In some sections of the ACM CCS (like in A.0), the classification is very accurate (using the 'Subjects:' classifiers) even on level 3 of the hierarchy. In this case, the topic 'A.0 *General* Subject: *Conference proceedings*' would only be similar with a value of 0.75 to itself $(3 * 0.25)$. To remedy such cases, we artificially add empty nodes (in the above example: A.0.0) into the topic hierarchy such that all topics described with 'Subjects:' are actually moved down into level 4. A manual inspection of the ACM CCS has shown that this is reasonable, since the topics described by 'Subjects:' are indeed very specific. In contrast, moving down topics on inner nodes (such as 'A.0 General') was not considered useful since most of the topics on these inner nodes are not very specific.

The final similarity matrix A in our case is a rather sparse block matrix because the ACM topics are sorted and the similarity between topics from different first levels (e.g., $A.$ vs. $B.$) is always zero.

In contrast to other popular metrics, such as the one proposed by Li et al. [6], our metrics handles the cases of different root topics in a more suitable way: The similarity of two topics at level 1 (e.g., B. 'Hardware' and E. 'Data') is only 0.25 while it is 1 for the Li metric. We consider this difference important to ensure that papers which are classified only very roughly (in level 1 topics) are generally ranked lower than very accurately classified papers(i.e, with level 4 topics). In all other respects both metrics behave rather similar as shown in Table 4, especially for the mostly used topics on the levels 3 and 4 (the most specific topics).

Table 4: Comparing Li & QFD for ACM CCS

Path length	Level of subsuming node	Li et al.	QFD
0	1	1	0.25
0	2	1	0.5
0	3	1	0.75
0	4	1	1
1	1	0.44	0.25
1	2	0.68	0.5
1	3	0.77	0.75
2	1	0.35	0.25
2	2	0.56	0.5
2	3	0.63	0.75
3	1	0.29	0.25
3	2	0.46	0.5
4	1	0.24	0.25
4	2	0.37	0.5
5	1	0.20	0.25
6	1	0.16	0.25

Besides our QFD similarity metrics, we also considered the Mahalonobis distance using the covariance matrix such that the similarity metric takes into account that topics used by many of the users do not have such a discriminating strength than topics used less often. This is the same motivation as for using the inverse document frequency in Information Retrieval [8]. However, an analysis of the ACM publications in DBLP has shown that no topic is used by more than 2% of the publications, so using a covariance matrix cannot be expected to have a large impact.

5 Summary

In this paper we proposed to use topic hierarchies for the design of user profiles for and provided two show cases, one based on the manually crafted ACM classification system and one based on topic hierarchies automatically created using our GrowBag approach. We also sketched a method to use such profiles together with their respective topic hierarchies to finding similar users or communities. Our future work will especially address open problems regarding the expressiveness of the derived models. We plan a large scale evaluation over different topic areas like medicine, where a comparison to heavily used and maintained classification systems as e.g., the MeSH ontology, is possible.

References

[1] J. Diederich and W.-T. Balke. The Semantic GrowBag Algorithm: Automatically Deriving Categorization Systems. In *Proc. of ECDL*, Sept. 2007.

[2] J. Diederich, W.-T. Balke, and U. Thaden. Demonstrating the Semantic GrowBag: Automatically Creating Topic Facets for FacetedDBLP. In *Proc. of JCDL*, 2007.

[3] S. Gauch, J. Chaffee, and A. Pretschner. Ontology-based personalized search and browsing. *Web Intell. Agent Systems*, 1(3/4), 2003.

[4] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, 1995.

[5] T. Iofciu and J. Diederich. Finding Communities of Practice from User Profiles Based On Folksonomies. In *Workshop on Building Technology Enhanced Learning solutions for Communities of Practice (TelCoPs)*, Crete, Greece, 2006.

[6] Y. Li, Z. Bandar, and S. McLean. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. *Transactions on Data and Knowledge Engineering*, 15(4):871–882, 2003.

[7] Z. Ma, G. Pant, and O. R. L. Sheng. Interest-based personalized search. *ACM Trans. Inf. Syst.*, 25(1), 2007.

[8] F. Menczer. Mapping the Semantics of Web Text and Links. *IEEE Internet Computing*, pages 27–36, 2005.

[9] Y. Petrakis, G. Koloniari, and E. Pitoura. On using histograms as routing indexes in peer-to-peer systems. In *Workshop on Databases, Information Systems, and Peer-to-Peer Computing*, pages 16–30, 2004.

[10] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[11] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *Proc. of SIGIR conference*, Berkeley, CA, USA, 1999.