# Semantically Faceted Navigation with Topic Pies

Tilman Deuschel
University of Applied Sciences Darmstadt
Department of Media
Dieburg, Germany
tilman.deuschel@h-da.de

Christian Greppmeier
University of Applied Sciences Darmstadt
Department of Comp. Science
Darmstadt, Germany
christian.greppmeier@stud.h-da.de

Bernhard G. Humm
University of Applied Sciences Darmstadt
Department of Comp. Science
Darmstadt, Germany
bernhard.humm@h-da.de

Wolfgang Stille
TU Darmstadt
University and State Library
Darmstadt, Germany
stille@ulb.tu-darmstadt.de

## ABSTRACT

Faceted search allows navigating through large collections along different dimensions in order to find relevant objects efficiently. Traditional faceted search systems often suffer from a lack of usability; furthermore facets are often static and independent from the search result set. In this paper, we present a dynamic semantic topical faceting approach. It uses a pie menu called *topic pie* that allows visualisation of facets and user interaction. Depending on the search query, the *topic pie* presents a set of topics and major topics which help the user to drill down the search result set to relevant objects efficiently as well as to browse exploratively through the collection. The underlying algorithm optimises the conflicting goals relevance and diversity while avoiding information overload. It reveals a good performance on large data sets. As our use-case, we chose literature research in scientific libraries. An evaluation shows major advantages of our approach compared to state-of-the-art faceted search techniques in nowadays library portals.

## Categories and Subject Descriptors

H.3.7 [**Information Storage and Retrieval**]: Digital Libraries; H.5.2 [**Information Interfaces and Presentation**]: Graphical user interfaces (GUI); I.5.3 [**Pattern Recognition**]: Clustering

## General Terms

Algorithms, Human Factors, Theory

## Keywords

Faceted Search, Mobile Devices, Visualisation, Topic Pies


*SEM '14* , September 04 - 05 2014, Leipzig, AA, Germany

ACM 978-1-4503-2927-9/14/09 ...$15.00
http://dx.doi.org/10.1145/2660517.2660522

## 1. INTRODUCTION

Faceted search can be seen as interactively exploring data according to different dimensions [7, 16]. This process may be iterative, which is called *drilling down* along one or more dimensions. In this paper[1], we focus on the example use case of literature research. Efficient faceted search for literature research is a complex engineering challenge that involves different tasks. A major task is to provide refinement options for the user based on structured knowledge for usually unstructured results derived from a query. Structured refinements help users to understand the categories the results belong to, to explore these categories, to disambiguate results by means of these, and to filter irrelevant ones [25]. They also reduce the number of query reformulations, thus save time, and may influence searchers' tactics [14].

Today's literature research platforms are expected to be an all-in-one search tool, where users tend to search by short queries in a brief search session, even in an academic setting [17]. Classic faceted search on these platforms usually has some major drawbacks: on the one hand, the facets shown are often impractical and not intuitive which prevents a majority of users from using them. For example, library-oriented knowledge is necessary to control the facets properly, or we are indeed able to select media types and library branches whereas the system lacks topical faceting. On the other hand, facets are usually static, which means that predefined facets are always visible regardless of the search results the query delivers, i.e., many of the shown facets are not useful in order to navigate through the search set and drill down the number of search results efficiently, like in the system presented in [25]. An eye tracking study revealed that facets are of major importance in the explorative search process [14]. Therefore we strongly believe that a successful design of faceted search mainly depends on both a selection of facets and the design of the user interface component for the faceted search.

With the growth of the mobile sector new literature research systems emerged to support mobile usage. The ap-

[1]This work has been funded by *LOEWE (Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz)* of the State of Hesse, Germany, under grant no. HA 321/12-11.

proach, described in [12] has been improved, as described in [21]. However, these approaches are not able to present hierarchies on the same screen and require frequent scrolling. Recent work of faceted search on mobile devices employs a pie menu with promising results regarding usability [11], as pie menus decrease the user's effort [22].

In our work, we combine an advanced faceting algorithm, that overcomes the drawbacks mentioned above, with a visualisation technique that optimally suits mobile devices and allows an intuitive navigation through the facets and the search result set even on small displays.

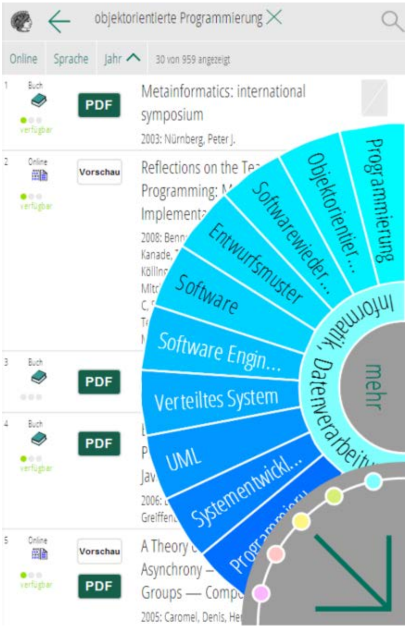


**Figure 1: Mobile research app using topic pies.**

## 2. TOPIC PIES

For this purpose we make use of a pie menu as an user interface component for the topical faceting, in the following called *topic pie*, similar to the visualisation recommended by Chang et al. [3]. Figure 1 shows the mobile app of our approach with the rotatable topic pie displayed. The topic pie is crossing the bleed, so that the text on the left is always horizontal and readable. We observed during the usability walkthrough, see Section 5.1, that this design decission also has positive impact on the pie's usage: most users wanted to read the labels that were not horizontally aligned and thus discovered by trial that the pie is rotatable. The topic pie also allows users to refine their search results. In our case, we use two layers of hierarchical information given by a taxonomy of topics. The outer ring of the topic pie assembles topics which themselves again contain a topically similar document collection each. The inner ring of the pie contains terms, which label an aggregation of several topics from the outer ring into more general topics, such that the hierarchical taxonomy is preserved, see Figure 2.

We use topic pies to implement topical faceting of search result sets, i.e., to explore and drill down the set of results corresponding to a given query according to topics. The benefit of showing two hierarchical levels at the same time is twofold: (1) it enables the user to navigate towards fine-grained topics, but also use a more coarse-grained level that

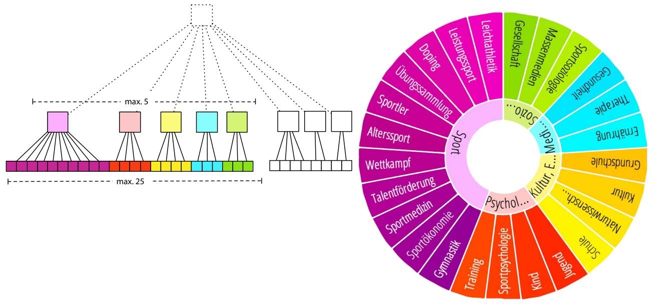


**Figure 2: The topic pie built from a taxonomy tree (topics in German)**

shows coherence of topics on a higher level, (2) it allows disambiguation of ambiguous search terms. A selection of a specific pie segment facets the search according to the corresponding topic and results in a smaller search result set. Then, a new topic pie is generated according to the new search result set. This allows users to iteratively refine their search. The topics used in our approach might be taken from existing metadata of the object collection, or be generated from fulltexts, for example using topic models [1].

The following example shows the drill down process of the faceted navigation for the initial search request "Java". This search request yields 1743 results and produces the topic pie shown in Figure 3 (left), which mainly displays Computer Science topics (shown in blue), but also others outside this subject area like Java, the island and Indonesia (in the yellow sector). In order to narrow down these results, the user may choose a specific topic from the outer ring, (e.g., Web Services) or the whole sector (e.g. Computer Science) by selecting an element from the inner ring. After choosing the domain Computer Science the search results are reduced to 645 and the resulting topic pie, shown in Figure 3 (middle), adapts immediately by presenting more topics from this domain. Finally Figure 3 (right) shows another drill down step: selecting the topic "Web Services" reduces the search to only 23 results.

## 3. OBJECTIVES AND CONSTRAINTS

Our objective is to support an intuitive and easy, but at the same time efficient and stringent way of literature research. This means that fast topic-related narrowing and classification of search results (topical faceting) is supported. Furthermore, the application aims at exploring and discovering (new) domains by giving access to literature without a specific search request. We formulate three requirements R1 to R3 to achieve this:

**R1.** *Topical faceting based on semantically relevant topics*: by "relevant" we mean that the chosen topics must refine the search, but do not produce empty result sets.

**R2.** *Semantically meaningful hierarchical ordering of topics*:

1. Topics in the inner ring are always broader than the ones in the outer ring. Topics in the inner ring are not in a broader, respectively narrower relationship with each other.

2. Broader topics allow disambiguation of narrower topics (e.g., Java as island and as programming language).

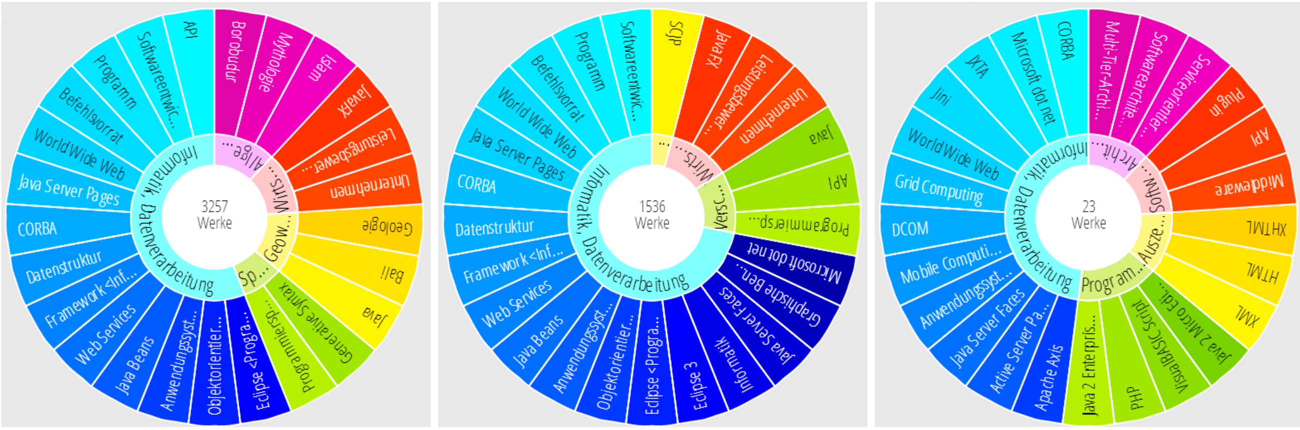


**Figure 3: Sequence of topic pies in iterative literature research: *(left)* after search input "Java", *(middle)* after selection of "Computer Science" in the inner ring, blue segment on the left hand side, and *(right)* after selection of Web Services on the wheel in the middle.**

**R3.** *Upper and lower bounds on the number of topics* in order to avoid information overflow and to guarantee sufficient diversity of topics:

1. Let $m$ be the number of topics in the outer ring. Then $m$ is bounded by a constant $C$, say $m \leq C$.

2. Let $n$ be the number of topics in the inner ring. Then $n$ is bounded by two constants, $\ell \leq n \leq u$.

We are now able to formulate the desired behaviour as optimisation problems $(O1)$ and $(O2)$ maximising two conflicting *optimisation criteria*:

1. $(O1)$ *Relevance:* maximise the sum of relevance scores of topics shown in the pie (each calculated by the belonging part of the document collection implied by the search result) subject to constraints $R3$ above.

2. $(O2)$ *Diversity:* maximise the number of major topics in the inner ring of the topic pie subject to constraints $R3$ above and another constraint ensuring the sum of relevance scores of topics exceeds a given lower bound.

# 4. AN ALGORITHM FOR THE GENERATION OF TOPIC PIES

## 4.1 Overview

Topic pies are a quite generic instrument to visualise and navigate through data collections. Depending on the application, available information, that might be used as input data for the algorithm, and concomitant the measures used in the approach might be very different. For this reason, we intentionally describe the algorithmic approach in a quite generic way and make mention of implementation details at the very end of this section.

Our algorithm that builds topic pies for semantically faceted navigation has the following inputs:

1. The result-set $\mathcal{R}$ of a search query on a collection $\mathcal{C}$ of objects including their metadata, e.g. documents from a library and their corresponding topics or index terms.

2. A hierarchical taxonomy $\mathcal{T}$ (a tree or a forest). In our case, we chose a taxonomy of topics, which we call a *topic tree*, see Figure 4.

3. A mapping function $p : \mathcal{C} \rightarrow \mathcal{T}$, which maps each object of the collection $\mathcal{C}$ to at least one element of the

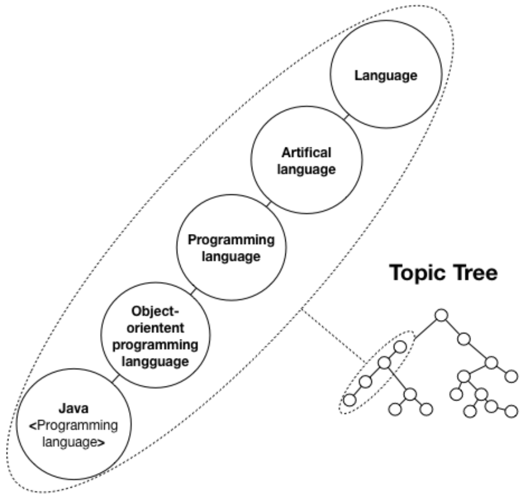


**Figure 4: Topic tree (right) and detailed view of a branch (left)**

The output is a topic pie that satisfies the requirements of Section 3. Our algorithm consists of the following steps which are given in detail in the following:

1. Identify and weight active nodes in the topic tree.

2. Cluster topics and identify major topics.

3. Shrink topic tree.

## 4.2 Step 1: Identify and weight active nodes in the topic tree

Initially, the algorithm collects all necessary metadata information from objects in the search result set $\mathcal{R}$ and assigns them each to one or more corresponding node(s) in the topic tree $\mathcal{T}$ by means of the mapping function $p$. When an object from $\mathcal{R}$ has been assigned to a node $t \in \mathcal{T}$, we call node $t$ *active*. Multiple assignments to one node generally yield a higher weight of this node.

Different approaches might be used in order to weight the nodes, i.e. determining the relevance of topics: One approach is to count the frequency of occurrence of topics in the search result set $\mathcal{R}$, as mentioned above. Another approach is to normalise the topical frequencies by the inverse document frequency *tf*idf*, based on the whole document collection or just on the search result set. Additionally, relevance measures from the index delivering the search result might be taken into account.

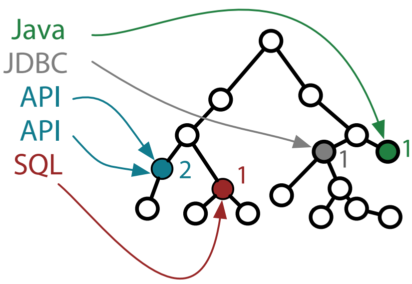


**Figure 5: Nodes in a topic tree get activated and weighted by mapping textual metadata information onto the tree.**

At the end of this process, we might have a high number of active nodes, possibly violating constraint R3. This might be fixed by several actions:

- iterative merge of two nodes in the same subtree: nodes are merged into predecessor nodes yielding a lower number of topics, whereas the merged nodes resemble more generic terms. Algorithm 1 illustrates a possible merging process by identifying nodes containing more than one active successor as merging candidates in a first step using an adapted depth-first search. Then, it iteratively uses these candidates to decrease the number of active topics until a specified number $C$ is reached. The merging process starts at the very bottom of the tree to ensure that the generalisation of topics starts with more specific topics rather than with more general ones.

  The merging process might also follow some objective such as preserving a best possible balance of relevance scores (weights) between the nodes, or merging subtrees containing the largest number of active nodes first, for example.

- Prune (sub-)trees, preferably by relevance (weight) of the (sub-)tree.

Note that the first action is conservatory with respect to the fact that each object from the result set $\mathcal{R}$ is contained in at least one active node from the topic tree, whereas initially chosen topics might be generalised by merging nodes. On the other hand, the second variant does not ensure containedness, but this might not be a problem for the application as only irrelevant topics are cut.

As a result of Step 1, we obtain a topic tree containing active and weighted nodes (see Figure 5). These are candidates for the outer ring of the topic pie.

**Algorithm 1: Cluster Topics**

1. Initialisation: **foreach** node $v \in \mathcal{T}$ **do**
   - **if** $v$ is inactive, **set** $a(v) := 0$, **else set** $a(v) := 1$, $A := \{v \in \mathcal{T} | a(v) = 1\}$
2. **call DFS**($\mathcal{T}$). During DFS traversal, **foreach** node $v \in \mathcal{T}$ DO :
   - save the distance $d(v)$ to the root node,
   - **set** $a(p(v)) := a(p(v)) + a(v)$ for the parental node $p(v)$ of $v$.
   - **if** $a(p(v)) > a(v)$ mark $p(v)$ as *merging candidate*.
3. **set** $\ell := \max\{d(v) | v \in \mathcal{T}\}$
4. **for** $i := \ell$ **to** 0 **do**
   - **foreach** merging candidate $v \in \mathcal{T}$ with $d(v) = i$ **do**
     - **set** $v$ active, **set** all successor nodes of $v$ inactive, update $A$
     - **if** $|A| \leq C$ **return**

## 4.3 Step 2: Cluster topics and identify major topics

In order to identify general topics for the inner ring of the topic pie, groups of active nodes from a branch are aggregated into nodes located at a higher level in the tree. This is done by iteratively identifying common ancestors of active nodes on a lowest possible layer within the tree until constraint R3 is satisfied. Again, Algorithm 1 is used to identify possible candidates, and the merge might be done according to a specific strategy, e.g. ensuring best possible balance of weight between the clusters. The results are illustrated in Figure 6 where active nodes (shown in green) are clustered according to their corresponding general topic nodes (marked as red).

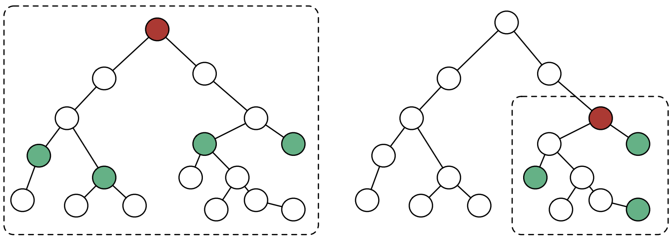

**Figure 6: Two topic clusters (marked by dotted rectangles) and their identified general topics (red) aggregated from topics (green).**

Depending on the properties of the taxonomy and the search result set, we cannot completely ensure feasibility of constraint R3. For example, this might happen in case of very precise search queries delivering only few results or even an empty set. Therefore, at this point, infeasibility has to be detected and handled accordingly, e.g. by not showing a topic pie at all for very small result sets.

On the other hand, the taxonomy might contain a large number of trees, and many root nodes have been activated as major topics. In this case, no further decrease of the number of major topics is possible by clustering (to cluster different topical trees indeed makes no sense as there is no relation between those). Then, ranking the trees and pruning them, might be the only option. In order to achieve this, clusters and their respective major topics are ranked by relevance. Again, there are different ranking approaches, e.g. via the sum or the maximum of its aggregated topic ranks. First, the lowest ranked clusters are eliminated as long as R3b is violated. Then, the topics in the remaining clusters are pruned. Again, there are different approaches. In the simplest approach, all lowest ranked topics (according to the topic rank from Step 1) are pruned as long as R3a is exceeded. If a prune operation violates constraint R3c (diversity), this is then skipped.

## 4.4 Step 3: Shrink topic trees

The last step consists of shrinking the topic tree into a 2-layer tree, which is used as basis for the topic pie as depicted in Figure 2. To achieve this, for all green nodes, we follow the paths in direction of the root node. As soon as we reach a red node, we connect the green node directly to this red node. All other nodes in the topic tree are dropped. Figure 7 illustrates the three steps of the algorithm by an example.

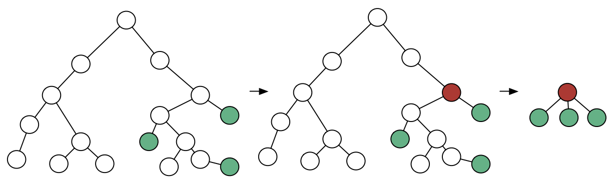

**Figure 7: Exemplary illustration of the algorithm: Outputs of Steps 1, 2, and 3.**

## 4.5 Customisation of the Algorithm

In order to implement the algorithm for a concrete scenario, it needs to be customised to the characteristics of the application. In particular, structure and quality of the underlying input data have to be considered carefully, which means that we have to give thought to questions such as: Is the present taxonomy usable as it is or is it necessary to pre-process it to meet the application's needs? Which measures are appropriate for the scenario? Is it better to merge or to prune tree nodes? Some of these questions may be answered in advance, some after experimentation with different settings.

In our application, we use index terms from the metadata of literature objects from a library system. These are mapped on a taxonomy from the GND Integrated Authoring File by the German National Library [23], which contains hierarchical orderings of topics. The GND taxonomy itself is highly fragmented with hundreds of separate trees; quite specific terms like *Web Services* are considered root nodes, which complicates the clustering process. In order to meet the requirements for our approach, we extend the GND taxonomy by another GND classification, which again maps sets of single trees to superordinate subject areas. By inserting this additional layer, we obtain a combined taxonomy consisting of around 20 separate trees.

Another issue might be performance. The document stock in our implementation contains around 10 million records; the GND taxonomy consists of over $400,000$ nodes. The required user response time should be less than one second for common literature searches. We met this challenging requirement by a number of data-specific performance optimisations such as denormalisation of database records and the implementation of custom data structures held in main memory, see also [9].

# 5. EVALUATION

The topic pie's design and quality of facets has been evaluated by qualitative and quantitative means: a) a usability walkthrough comparing the topic pie application with a competitor to gather quantitative data and to observe user behaviour, b) an odd candidate test, to explore the homogeneity of the refinement recommendations and c) a usability questionnaire to discover how the users experienced the topic pie during the usability walkthrough. All tests have been conducted with 14 subjects, half of them were students, the other half were librarians.

## 5.1 Usability Walkthrough

### 5.1.1 Design

In a standard usability walkthrough the subject is accompanied by the test manager during the testing period solving realistic tasks [20]. Each subject solved in total four tasks on both systems and on both device classes. The systems are: 1) the fully functional topic pie system and 2) a modern literature research system, that employs a collapsible faceted search with flatter facets than system 1). The devices are: a) a Full HD laptop and b) an 10" tablet. To avoid a bias by the learning effect, the tasks where randomly assigned to a system-device combination.

The dependent measures are *obtaining*, *utilisation* and *yield*. The value obtaining shows how many relevant literature have been obtained per minute by the subject and is thus the inversion of task completion time from which interruptions like questions have been substracted. The value utilisation shows how often the faceted search has been utilised per minute. A high utilisation of the faceted search indicates the curiosity of the user, as the tasks could be solved without interacting with the faceted search. Yield is the quotient of number of relevant literature found by using the faceted search divided by the total number of relevant literature found. A high yield indicates the faceted search's effectivity.

### 5.1.2 Results

The median of the obtaining measure shows that the subjects performed slightly faster with system 2) than with system 1) (Figure 8 (a)). During the usability walkthrough we could observe, that the topic pie served to explore possibilities, while the faceted search of the competitor system was often unnoticed thus more time was spend using system 1) than using system 2) because of curiosity. It shows that the topic pie has been used more frequently than the faceted search of the competitor system (Figure 8 (b)). The median of the yield measure shows that the topic pie application outperforms the competitor system on desktops (Figure 8 (c)), while the yield on tablets is similar. We can conclude that the topic pie encourages users to engage and fosters the discovery of literature and it outperforms the faceted search of modern literature research portals on desktops.

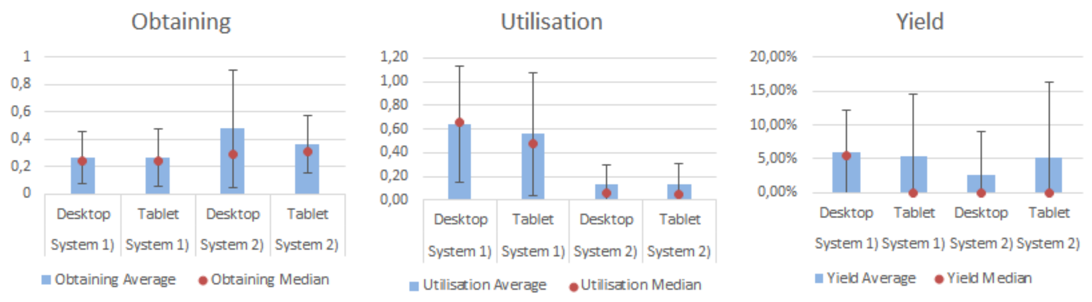


**Figure 8: Usability walkthrough results (a) Obtaining, (b) Utilisation and (c) Yield.**

## 5.2 Usability Questionnaire

### 5.2.1 Design

The questions are derived from the ISO norm for designing interactive dialogues [10] and the *Questionnaire For User Interaction Satisfaction* (QUIS) [8]. QUIS employs an uneven Likert scale and one dimensional questions. It is discussed in [19] that this can cause a misunderstanding of the subjects, as there is no center but 50 % compliance. [19] also discusses the problem of providing too many choices on the scale possible. We addressed these issues by designing a questionnaire covering the most important aspects of utility, ease of use, learnability and satisfaction introduced by [10] harnessing a six point Likert scale and questions that don't bear an expectation towards the subject.

### 5.2.2 Results

The results show that the topic pie is fun to use as well as very easy, time-saving, user friendly and self-explanatory. Learning of the topic pie is percepted as easy and the mastery happens immediately. The subjects think that both a frequent and an infrequent user would rather like the topic pie and they would recommend the application to a friend. When using the topic pie they also perceive it as usefull and the recommendations are experienced as satisfying though the topic pie's recommendation are experienced as rather unpredictable and the subjects feel rather less control.

## 5.3 Odd Candidate Test

### 5.3.1 Design

We use an odd candidate test similar to the one described in [3]. 18 topic pies have been preselected twelve of which contained a prepared "faulty" recommendation in addition to 24 further recommendations to a given search term provided by the algorithm. The remaining six topic pies did not contain a "faulty" recommendation. The subjects were told, that some topic pies do not contain a faulty recommendation and some do only contain one faulty recommendation. They were asked to mark the flawless topic pies or write down the recommendation of which they think it is faulty.

### 5.3.2 Results

The success rate of identifying the flawless topic pies is 79% with a standard deviation of 30, similar to the success rate of identifying the faulty topic pie which is 84% with a standard deviation of 15. One faulty topic pie had a success rate of only 20%. The error was obviously too hard to detect. When its data set is excluded, the success rate and standard deviation is very similar. We conclude that the recommendations are homogeneous and meaningful to the given search term.

# 6. RELATED WORK

## 6.1 Faceted search on Mobile Devices

McGrath et al.[16] present a multi-faceted search interface to improve the searcher access to library collections of movies and programs. For the same use case, Wilson and Schraefel present a directional column-faceted browser that provides information about relationships within an digital library [24]. Directional column-facets are horizontally presented facets that filter only to the right to allow inter-facet relationships. Both works give a good overview about the use of facets in literature research systems in libraries, but do not specifically address the mobile sector.

Karlson et al.[12] present in the beginning of the smartphone age *FaThumb*, a multi-faceted search interface for mobile devices. A total number of nine different facets, exploiting multiple faceted taxonomies, is possible because the user has to navigate using the phone's number keys to drill down. The presented work is based on a system-centered approach. More recent approaches utilise a user-centered approach. Kajiyama and Satoh [11] present a multi-faceted search for mobile devices to search the *Appstore* for apps. The facets are layouted as rings, that can be turned to select the wanted refinement. There is no combination with search terms possible, because this system focuses on search by touch. This system provides no hierarchical facets, so no down-drilling is possible. Lambeck et al. present in [15] a multi-faceted search user interface component for retrieving media items in a car manufacturer's digital archive. The search is similar to the presented topic pie but it can be used collaboratively on multi touch tables and exploits multi-facets in sections of one ring, which results in arced lists that have to be scrolled. This approach is rather inappropriate to harness on mobile devices, due to the limited screen real estate and the comfortable possibility to rotate user interface components.

## 6.2 Structuring and Visualisation of Document Collections

Zhang et al.[25] describe an environment to support literature research based on a writing's references and the respective cite notes. The categorical *ACM* knowledge structure

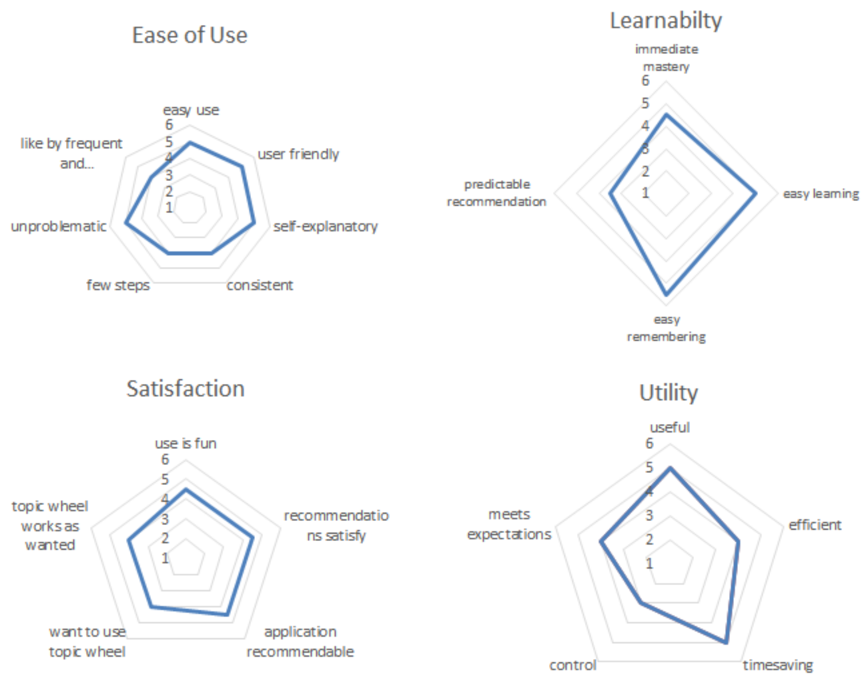


**Figure 9: Questionnaire results (a) Ease of use (b) Learnability (c) Satisfaction (d) Utility.**

has been used to structure the literature found. Kraker et al. [13] propose a domain visualisation called Head Start based on altmetrics data rather than citation data. It is intended for scholars who want to get an overview of a research field. They use readership co-occurrence as an indicator of scholarly activity. A related technique that generates hierarchical cluster trees and visualises it by means of points in 2D space in order to explore document collections visually is presented by [18]. Similar to our approach, the suggested procedure allows the user to control the degree of detail while exploring a document collection. However, visualisations from this approach might become quickly complex or confusing with large heterogeneous collections as an input. Another technique that makes use of topic hierarchies is presented by [6], although the application context is slightly different: They propose a so-called topic rose tree in order to support scalable visual analysis and exploration of text corpora. As in our case, the tree supports variants of aggregation methods in order to control the degree of granularity, but the main limitation of this technique is also the lack of scalability while displaying very large document collections.

SolarMap[2] makes use of an interactive topic ring showing multiple facets at the same time. Furthermore, it renders the evolution of topic clusters over time and integrates layout optimisation algorithms. The approach is able to show a lot of information in one diagram, which makes it quite powerful and suitable to deeply explore specific research topics. At the same time, information density is high, which requires some practicing with the tool and limits its usage on mobile devices where limited space is available.

PaperVis[4] is a tool that helps researchers reviewing literature by showing relations between documents, e.g. the citation-reference structure in a collection of articles. Furthermore, it categorises papers into semantically meaningful hierarchies aiming at facilitation of literature exploration. One graphical element used in this approach is quite similar to our topic pies. However, the approach is suitable for a set of documents rather than for large collections.

TextWheel [5] is a keyword wheel similar to a text cloud to display keywords of news streams and the links between them. DocuBurst [**?**] uses a radial, space-filling layout to visualise document content by using human-created structure from lexical databases.

## 7. CONCLUSION AND FUTURE WORK

Faceted search is state-of-the-art in library portals but usability is still poor. In this paper, we have presented a dynamic semantic topical faceting approach with a pie menu called *topic pie*. The underlying algorithm optimises the conflicting goals of relevance and diversity while avoiding information overload. The algorithm uses ranking and graph traversal over a taxonomy and exhibits good performance. An evaluation with users shows major advantages of our approach compared to state-of-the-art faceted search techniques in library portals. Because of the promising results of the prototype implementation we plan to improve the topic pie, based on findings from intensified evaluation effort, and put it into productive use in German state libraries.

## 8. REFERENCES


[1] D. M. Blei and J. D. Lafferty. Topic Models. In *Text Mining: Classification, Clustering, and Applications*, pages 71–89. 2009.

[2] N. Cao, D. Gotz, J. Sun, Y.-R. Lin, and H. Qu. Solarmap: Multifaceted visual analytics for topic

exploration. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 101–110, Dec 2011.

[3] J. Chang, S. Gerrish, C. Wang, and D. M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296, 2009.

[4] J. K. Chou and C. K. Yang. Papervis: Literature review made easy. *Computer Graphics Forum*, 30(3):721–730, 2011.

[5] W. Cui, H. Qu, H. Zhou, W. Zhang, and S. Skiena. Watch the story unfold with textwheel: Visualization of large-scale news streams. *ACM Trans. Intell. Syst. Technol.*, 3(2):20:1–20:17, Feb. 2012.

[6] W. Dou, L. Yu, X. Wang, Z. Ma, and W. Ribarsky. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2002–2011, Dec 2013.

[7] J. English, M. Hearst, R. Sinha, K. Swearingen, and K.-P. Yee. Hierarchical faceted metadata in site search interfaces. In L. Terveen and D. Wixon, editors, *CHI '02 extended abstracts*, pages 628–639, New York and NY and USA, 2002. ACM Press.

[8] B. Harper, L. Slaughter, and K. Norman. Questionnaire administration via the www: A validation and reliability study for a user satisfaction questionnaire. In S. Lobodzinski and I. Tomek, editors, *Proceedings of WebNet 97*. AACE, 1997.

[9] T. Heuss, B. Humm, T. Deuschel, T. Fröhlich, T. Herth, and O. Mitesser. Semantically guided, situation-aware literature research. In *Workshop on User Interaction built on Library Linked Data (UILLD 2013), 79th World Library and Information Conference, Singapore*, 2013.

[10] ISO. DIN EN ISO 9241-110 Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung (ISO 9241-110:2006); Deutsche Fassung EN ISO 9241-110:2006, 2006.

[11] T. Kajiyama and S. Satoh. An Application Search Interface Including Sense-related Search Facets. In J. Jose, K. van Rijsbergen, M. Kankanhalli, S. Rueger, and R. Manmatha, editors, *International Conference*, pages 463–466, New York and NY and USA, 2014. ACM Press.

[12] A. K. Karlson, G. G. Robertson, D. C. Robbins, M. P. Czerwinski, and G. R. Smith. FaThumb: a facet-based interface for mobile search. In G. Olson and R. Jeffries, editors, *the SIGCHI conference*, pages 711–720, 2006.

[13] P. Kraker, K. Jack, C. Schlögl, C. Trattner, and S. Lindstaedt. Head Start: Improving Academic Literature Search with Overview Visualizations based on Readership Statistics. In *Web Science 2013*, 2013.

[14] B. Kules, R. Capra, M. Banta, and T. Sierra. What do exploratory searchers look at in a faceted search interface? In F. Heath, M. L. Rice-Lively, and R. Furuta, editors, *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 313–322, New York and NY and USA, 2009. ACM Press.

[15] C. Lambeck, J. Wojdziak, and R. Groh. Facet lens: local exploration and discovery in globally faceted data sets. In J.-B. Martens and P. Markopoulos, editors, *Procedings of the Second Conference on Creativity and Innovation in Design*, pages 85–88, New York and NY and USA, 2011. ACM Press.

[16] K. McGrath, B. Kules, and C. Fitzpatrick. FRBR and facets provide flexible, work-centric access to items in library collections. In G. Newton, M. Wright, and L. Cassel, editors, *Proceeding of the 11th annual international ACM/IEEE joint conference*, pages 49–52, New York and NY and USA, 2011. ACM Press.

[17] X. Niu and B. M. Hemminger. Beyond text querying and ranking list: how people are searching through faceted catalogs in two library environments. In C. Marshall, E. G. Toms, and A. Grove, editors, *Proceedings of the 73rd ASIS{&}T Annual Meeting on Navigating Streams in an Information Ecosystem*, Silver Springs and MD and USA, 2010. American Society for Information Science.

[18] F. Paulovich and R. Minghim. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1229–1236, Nov 2008.

[19] R. Porst. *Fragebogen: Ein Arbeitsbuch*. Studienskripten zur Soziologie. VS-Verl., Wiesbaden, 13, 4. aufl edition, 2014.

[20] M. Richter and M. Flückiger. *Usability Engineering kompakt: Benutzbare Software gezielt entwickeln*. Spektrum, Akad. Verl., Heidelberg and Germany, 2. aufl edition, 2010.

[21] M. Schneider, A. Scherp, and J. Hunz. A comparative user study of faceted search in large data hierarchies on mobile devices. In M. Kranz, K. Synnes, S. Boring, and K. van Laerhoven, editors, *12th International Conference on Mobile and Ubiquitous Multimedia*, pages 1–10, New York and NY and USA, 2013. ACM Press.

[22] B. Sears and B. Shneiderman. Split menus: effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction*, 1(1):27–51, 1994.

[23] B. Wiechmann. German national library: Integrated authoring file, http://www.dnb.de/en/standardisierung/gnd/gnd_node.html, visited on 16th of June 2014.

[24] M. L. Wilson and M. Schraefel. A longitudinal study of exploratory and keyword search. In R. Larsen, A. Paepcke, J. Borbinha, and M. Naaman, editors, *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 52–56, New York and NY and USA, 2008. ACM Press.

[25] X. Zhang, Y. Qu, C. L. Giles, and P. Song. CiteSense: supporting sensemaking of research literature. In M. Czerwinski, A. Lund, and D. Tan, editors, *Proceeding of the twenty-sixth annual CHI conference*, pages 677–680, 2008.