Exploring and Summarizing Document Colletions with Multiple Coordinated Views

Cecilia di Sciascio
Know-Center GmbH
Graz, Austria
cdisciascio@know-center.at

Lukas Mayr
Graz University of Technology
Graz, Austria
lukas.mayr@student.tugraz.at

Eduardo Veas
Know-Center GmbH
Graz, Austria
eveas@know-center.at

ABSTRACT
Knowledge work such as summarizing related research in preparation for writing, typically requires the extraction of useful information from scientific literature. Nowadays the primary source of information for researchers comes from electronic documents available on the Web, accessible through general and academic search engines such as Google Scholar or IEEE Xplore. Yet, the vast amount of resources makes retrieving only the most relevant results a difficult task. As a consequence, researchers are often confronted with loads of low-quality or irrelevant content. To address this issue we introduce a novel system, which combines a rich, interactive Web-based user interface and different visualization approaches. This system enables researchers to identify key phrases matching current information needs and spot potentially relevant literature within hierarchical document collections. The chosen context was the collection and summarization of related work in preparation for scientific writing, thus the system supports features such as bibliography and citation management, document metadata extraction and a text editor. This paper introduces the design rationale and components of the PaperViz. Moreover, we report the insights gathered in a formative design study addressing usability.

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation (e.g., HCI): User Interfaces—Graphical user interfaces (GUI)

Author Keywords
exploration, discovery, document collection exploration; paper writing application; graph-based visualization

INTRODUCTION
Due to the massive amount of digital data available on the Web and its steady growth, it is usually not hard to find scientific literature for a specific research field. Also, numerous search engines, such as Google Scholar, IEEE Xplore or Pubmed are specialized in retrieval of academic literature. But despite these means, identifying valuable resources is still challenging.

ESIDA'17, March 13 2017, Limassol, Cyprus

DOI: http://dx.doi.org/10.1145/3038462.3038471

To point out some common issues researchers face nowadays, it is first of all important to take a look at the process of finding and evaluating research materials. After defining a problem of interest, the researcher has to: (i) define search expressions, (ii) browse the Web with a general or academic search engine, (iii) review and evaluate the retrieved literature, (iv) derive new search expressions (key phrases, authors, journals, etc.) from these documents, and (v) start over at step *(ii)*, trying different combinations of search expressions [3].

Less iterations are needed the the researcher only needs a small set of documents and is able to instantly formulate a precise search query. However, in practice this is rarely the case [17]. At the beginning of a knowledge acquisition process, researchers may not have a strategy of what to search for and what facets should be covered in their papers. As a consequence, search expressions are too general and result in huge lists of low-quality and irrelevant content. Spotting valuable material turns then time consuming [3]. In particular, search results from academic engines are typically provided as a list of abstracts that the researcher has to read or scan to determine the relevance of a document for the topic of interest. Furthermore, they need to identify suitable key phrases for describing target papers. This step is crucial for optimizing the search results in further iterations. Thus, finding relevant information is not only a simple lookup task. Researchers need to learn from search results and develop a search strategy other than trial-and-error in order to refer to that information later. Search tasks with learning or investigative purposes are often regarded as exploratory search. Exploratory search requires for the information seeker to spend time viewing, comparing and making qualitative judgments about sets of objects [15].

In addition, researchers have to also take care of managing found resources for later detailed exploration. Most exploratory search systems focus on the process of finding content and learning along the way, but rarely provide enough support to fill the gap between this initial stage and production tasks. In other words, after the researcher has collected a large amount of resources, probably spanning more than a single topic and throughout several search sessions, at some point they need to make active use of these resources for production, i.e. writing purposes. In short, the challenges faced by researchers can be summarized as follows: (i) finding suitable keywords and search expressions, (ii) quickly identifying relevant resources within large repositories, (iii) organizing documents for later inspection, and (iv) leveraging collected resources for writing tasks.

To the extent of our knowledge, there is no system to date capable of addressing all these issues. In this paper we introduce a novel interactive tool called PaperViz, designed to support exploration tasks at production stages. To do so, PaperViz combines different visualization techniques and multiple coordinated views that, on one hand, allow the user to discover possible search tracks in terms of key phrases and, on the other hand, provide information at three levels of granularity: at collection, document and intra-document level. Finally, we report on a preliminary design study where we validate our design decisions and identify further necessary improvements.

RELATED WORK

Our work aims to support the discovery of useful resources distributed throughout several collections, possibly covering a variety of topics. The most popular approaches to visualize document collections at different levels of granularity can be roughly classified [9] as: (i) **collection level**: aim to provide an overview of the content of collections as a whole, (ii) **document level**: to visualize similarities and differences between documents, and (iii) **intra-document level**: to illustrate the internal structure of a document.

Visualizations at collection level mostly target relationships between documents and topics. In many cases, the underlying topic models are generated through probabilistic approaches like latent Dirichlet allocation (LDA) [4]. Visual techniques for topic collection relationships include clustering metaphors, e.g. galaxies [2] or topographic maps [7]; parallel coordinates [8] and graph-based layouts [12]. Tag clouds are a simple but popular option for overview and filtering of large text corpora.

At document-level, visualizations are intended to convey document attributes, such as similarity with respect to a search query or to other documents. Users query (query-focused) or explore (browsing-focused) their collections with interactive visual search interfaces. Typically, documents retrieved by a search engine are presented as ordered lists. The disadvantage is that lists force a sequential scanning. Moreover, [19] highlights the importance of providing relevance score explanations because they encourage users to explore beyond the first two results. Also, users prefer bars to illustrate relevance scores over numbers or the absence of graphical explanations.

Spatial similarity-preserving techniques convey inter-document or document-query relationship strength as a complement ordered lists. VIBE allows the user to set points of interest (POI), represented by unique icons, such that proximity to other objects indicates a strong influence of the POI [1]. The Apolo system allows for exploration within a citation network projected around an article of reference [6]. Other examples employ force-directed or alternative layouts [10, 16] to place topically similar document near each other.

At intra-document level, the scope of the visualization is generally the internal structure of a document, e.g. frequency and distribution of keywords. TileBars [13] make an efficient use of space to encode relative term frequency and distribution with shaded blocks. WordTree [11] describes likelihood of word sequences through a branching structure similar to decision trees. PhraseNet [20] displays a node-link diagram emphasizing connections among concepts within a text corpus like a paper, book or poem.

In this work we combine techniques to convey information at all levels: (i) a tag cloud for collection and document overview, (ii) a graph-based visualization for exploration of relevance relationships between key phrases and collection items, and (iii) tile bars for single-page descriptions. With the multiple perspectives we aim to support users in finding suitable resources for writing tasks, all in one integrated tool.

THE PAPERVIZ SYSTEM

PaperViz[1] is an interactive intended to assist researchers in finding suitable resources that match the current interests, reflecting on their content, building associations, and summarizing the outcomes in a document draft. It does so by enabling navigation, inspection and citation within the tool. PaperViz works on a repository of collections and documents gathered by the researcher in the course of (probably several) past search sessions, obtained from user's Mendeley account via the Mendeley's API[2].

Navigating, exploring and managing a large collection of documents in a Web browser usually results in several open windows or tabs at the same time. PaperViz provides a fluid navigation by distribute the UI components in one centralized screen comprising multiple coordinated views (MCVs) with a flexible layout. Collections, subcollections and documents are shown in a hierarchical structure with a tree view (Figure 1.A), which provides an overview of all available resources and primarily serves as a navigation control. Clicking an item in the tree updates the other views accordingly. A tag cloud (Figure 1.B) presents the 20 most frequent terms in a document or collection selected in the tree view, with font size and value (blue shading) encoding frequency. The tag cloud is intended to give a rough idea of the underlying topics before further inspection. Users drill into collections or documents by clicking on tags, thus extending their information needs. Clicked tags are replicated in the *Key Phrases* area (Figure 1.E). This area represents a "bag" that gathers all phrases of interest. From this bag, the user can pick a few phrases or terms to parametrize the graph visualization. Alternatively, the user can manually add a phrase by typing in the text field in the header of the Key Phrases box.

Inspecting resources in-depth is supported with the graph visualization (Figure 1.G). It enables inspection at at collection, document and intra-document level (see Section 3.1). The breadcrumbs atop (Figure 1.F) indicate the level in the hierarchy for the current graph view. The graph visualization, the breadcrumbs and the tree view function as MCVs, so that changes of state in one of them are reflected in the other two. Hence, clicking on a collection or document, regardless of which control is used, triggers the following actions:

- The corresponding tree view item is highlighted with a dark-grey background.
- The breadcrumbs reveal the hierarchical path of the item.

[1]Demo video available at https://youtu.be/td9ENIIIZGI

[2]http://dev.mendeley.com/methods/

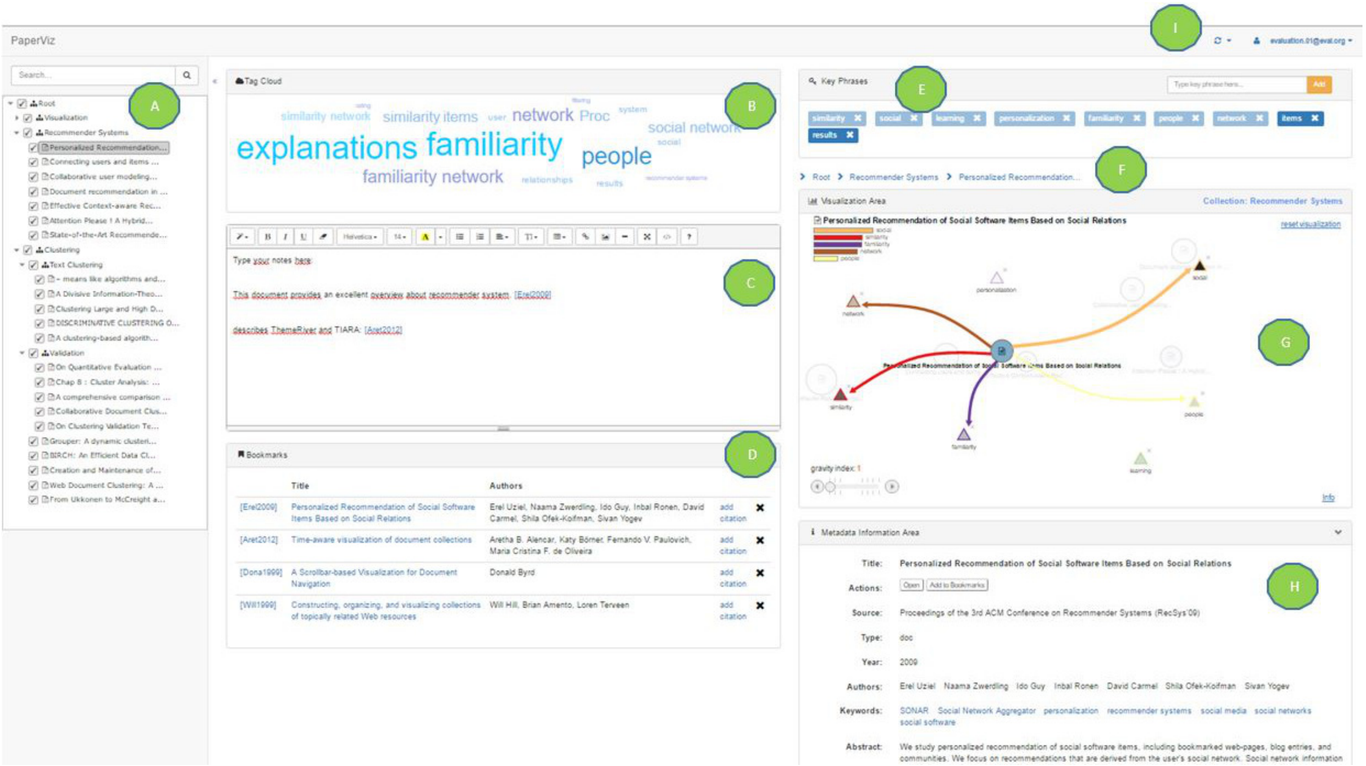


Figure 1: Screenshot of PaperViz UI showing its main components.

- If the selected item is a collection, the visualization area is populated with the documents therein. If it is a document, the visualization displays its tile bars widget. Other items at the same level (siblings) are also visible.
- The tag cloud of the selected item is shown.
- Additional information for the selected item is displayed in the metadata information area.

The *Metadata information area* (Figure 1.H) allows the user to open a document's PDF file in a new window, export its keywords to the Key Phrase area, or bookmark it. Bookmarked documents in panel (Figure 1.D) can be added as references into the editor (Figure 1.C), in order to link notes with the corresponding resources.

Graph-based Visualization

The graph-based visualization is the key component for interactive sensemaking of hierarchical document collections in PaperViz. It provides an overview of collections, documents and single document pages, enabled by mappings from textual to graphic representations. The goal is to help users readily spot documents and sections thereof that likely bear useful information for their writing purposes, by capitalizing pre-attentive visual patterns prior to detailed reading. In contrast to other visualization approaches, where the structure of the information is predefined by the data, PaperViz relies on a user-driven model that allows researchers to shape the visualization based on their interests.

PaperViz allows users to span their own information space by interactively formulating a search query. In this context a search query consists of a set of key phrases and their coordinates within a spatial layout. Key phrases can be added via drag-and-drop from the key phrase box into an arbitrary position in the visualization area. By adding at least two key phrases, the sub-collections and documents within the currently selected collection are displayed. Documents and collection nodes are then organized within the layout basing on the search query. The position of a document thereby reveals its strength with respect to the query terms. So, if a document strongly relates to a key phrase, the distance between them will be relatively short compared to those with a weak relationship. This approach enables the researcher to associate relative positions of documents to their content. Key phrase can be removed from the visualization and enabled again in the key phrase box by clicking the cross icon.

Furthermore, PaperViz provides complementary mini-visualizations depicting keyword distribution throughout pages in a document. Thus, by using their visual pre-attentive skills, researchers can quickly examine their collections.

Spatial Layout

In order to feed the graphical representations, PaperViz leverages the Sensium[3] text mining engine as Software as a Service (SaaS). Relevance scores are obtained by computing *term frequency – inverse document frequency* (*TF-IDF*) not only for each document, but also for every collection individually. Thus, these scores indicate relationship strengths between key phrases and text blocks. The text mining engine

[3] https://www.sensium.io/

thereby assigns scores ranging from 0 (no relationship) to 1 (the strongest). So comparing key phrases with documents and collections yields a matrix of scores. Each cell of this matrix represents the relationship strength between a key phrase and a certain item of a collection (document or sub collection).

Given matrix $S^{|CI|\times|KP|}$, where CI is the set of all collection items (either a collection or document) and KP is the set of all extracted key phrases, element $s_{ij} \in S$ indicates the strength between collection item i and key phrase j. We can then compute the diagonal matrix $D^{|CI|\times|CI|}$, such that each element $d_{ii} \in D = \frac{1}{\sum_{j\in KP}|s_{ij}|}$ represents the $L1$ norm (or least absolute deviation) of row i in S. Multiplying D by S produces a $|CI| \times |KP|$ matrix where every cell equals the normalized value for the original s_{ij}. In turn, matrix $Q^{|KP|\times 2}$ contains the coordinates for each key phrase j dropped by the user in the visualization area, such that $q_j = (x_j, y_j)$. Then, matrix $P^{|CI|\times 2}$ with X-Y coordinates for collection item nodes is obtained by multiplying normalized strength values with the current locations of selected key phrases in KP, as shown in Formula 1. In other words, the position of a node is defined by the set of forces acting upon it, whereby the magnitudes of these forces are represented by the corresponding row vector in S and the directions are defined by the coordinates of the key phrases.

$$P = DS^{\lambda_g}Q \tag{1}$$

λ_g is a gravity factor applied over score matrix S, so that larger values make document and collection nodes float towards the key phrase with the strongest relationship. This value is set to 1 by default and can be tuned via the gravity index slider (bottom-left corner).

To avoid overlaps, given two collection item nodes, we determine the occurrence of an overlap if the difference between the sum of their radii and their Euclidean distance is positive, i.e. $diff = radii - dist > 0$. In this case their centers need to be re-calculated. Hence, x-y offsets are obtained as follows:

$$o_x = \left(\frac{dist}{2diff}\right)\cdot(x_2 - x_1) \qquad o_y = \left(\frac{dist}{2diff}\right)\cdot(y_2 - y_1)$$

New central points are computed by subtracting x-y offsets to the original coordinates of the first node and adding them to the coordinates of the second one. The procedure is repeated for every pair of nodes until no overlap is detected.

Visual Encoding

After position, the encoding of several pieces of information relies on the following visual channels:

Shape. Geometric shapes are used to differentiate node types. Key phrases are visualized as triangles, whereas collection items are represented by circles. To distinguish collections from documents, we use sitemap and text icons, respectively (same as in the tree view).

Color. Each triangle has a unique border color assigned, which serves as an exclusive identifier. The qualitative color palette was generated by ColorBrewer[4].

Size. The diameter of a circle represents the size of a collection or document in terms of number of contained documents or number of pages, respectively. It is computed by the empirical formula $r_j = \log(size_{CI_j}) \cdot 7 + 5$, where r_j is the radius of the circle representing item j (in pixels). A linear function would yield cumbersome results, as the size of documents and collections can differ significantly.

Saturation. While proximity indicates the relationship strength between a key phrase and a given collection item, overall relevance of an item is encoded through the level of saturation (alpha value in the background color). In the case of key-phrase nodes, saturation is used to emphasize which query terms are the most relevant for the collections and documents in display. The darker the background, the higher its relevance. In order to maximize distinctiveness, the scale is normalized to values between 0 and the highest relevance score. Thus, the most relevant key phrase is always rendered with a black background. In contrast to the unique color, the intensity of a triangle changes when: *(i)* an even more relevant key phrase is added to the visualization, *(ii)* the most relevant key phrase is removed, or *(iii)* another collection is selected by the user. Similarly, the saturation of a circle denotes the relevance of a collection item with respect to all search terms.

Interactions and Details-on-demand

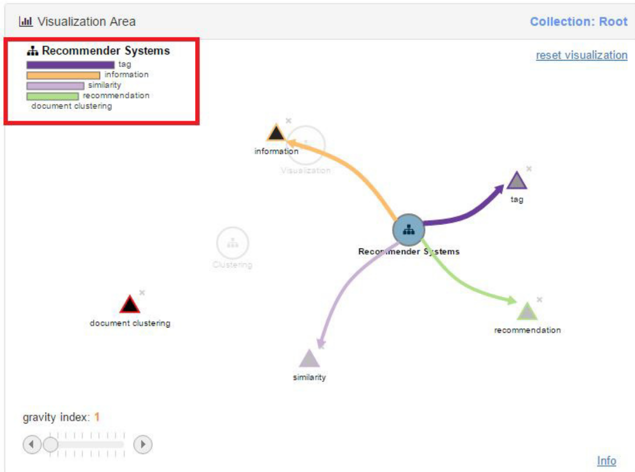
Mouse over and click events enable detailed inspections in the grap. Mouse over any node shows additional information regarding this item in form of a bar chart and connecting lines, while other nodes are dimmed. The trigered action after clicking on a circle depends on its type. By clicking on a collection circle, the user navigates into a deeper level in the hierarchy, meaning that contained sub collections and documents are now displayed. Clicking on a document node opens a tile-bar visualization, revealing information about single pages of the document.

Bar Charts

Sometimes it is not possible to identify the decisive forces for the placement of documents and collections. Hovering with the mouse pointer over a circle or triangle reveals ordered bar charts at the top-left corner with additional relevance information. Bar widths convey relationship strengths between key phrases and documents or collections. Figure 2a shows bar charts for a hovered collection (‘Recommender Systems’), such that bar colors match the corresponding key phrases. By looking at the bar chart, one can easily identify ‘tag’ as the key phrase with the strongest relationship, followed by ‘information’, ‘similarity’ and ‘recommendation’. The key phrase ‘document clustering’ is not relevant to this collection and therefore no bar is displayed.

In turn, hovering over a key phrase triangle shows the scores for the most related sub-collections and documents. Figure 2b illustrates a bar chart for the term ‘similarity’. As collections

[4] http://colorbrewer2.org/



(a) Hovered collection

(b) Hovered key phrase

Figure 2: Edges and informative bar charts provide details-on-demand by hovering on the different node types.

and documents do not have a unique color, all bars' backgrounds match the color assigned to the hovered key phrase. The icon next to each bar indicates whether the object is a collection or document.

Connecting Edges

Drawing connecting lines between elements of the visualization is another way to visualize relationship strength for a hovered element (Figure 2a and 2b). In contrast to bar charts, edges enable users to quickly spot the positions of related elements. The thickness of an edge thereby indicates the strength of the relationship between two objects. One could also state that these edges illustrate the forces acting upon the hovered object. To reduce the occurrence of overlaps, lines are rendered curved instead of straight.

Tile Bars

By clicking on a document circle, a tile-bar visualization is displayed, showing the distribution of key phrases throughout

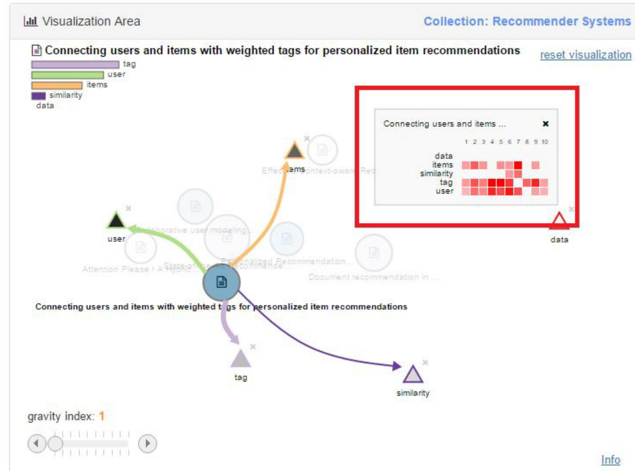


Figure 3: Tile bars showing distribution of key phrases throughout the pages of a document.

the pages of the selected document. This allows the researcher to quickly identify pages of interest before opening a document file. Figure 3 illustrates a tile-bar mini-visualization for the article 'Connecting users and items with weighted tags for personalized item recommendations'. Columns represent page numbers, whereas rows are words in the search query. Thef intensity of a single cell indicates the frequency of a key phrase in within a specific page. So if the researcher is interested in the term 'similarity' he can quickly identify page 6 and 7 as potentially relevant. By clicking on the corresponding cell, the article is loaded in a separate window and automatically scrolled to the desired page , which is supported by almost every modern PDF viewer. Highlighting a specific key phrase within the PDF file is currently not provided.

Resource Management

Resource management elements are GUI components that directly support production activities, namely: metadata information, bookmark list and text editor areas.

Metadata information area (MDA)

The information presented in this panel varies depending on the type of selected item. For a collection only title, number of contained sub-collections and documents are displayed. For a document the displayed information includes: (i) title, (ii) source, (iii) year of publication, (iv) list of authors, (v) author-defined keywords, (vi) abstract. Additionally, this area contains two buttons. By clicking on 'open' the PDF file of the document is loaded in a separate browser window/tab, while the 'Add to Bookmarks' button is used to add the selected document to the bookmarks list.

Bookmarks list

This list contains five columns: (i) *Abbreviation:* acronym (first 4 letters of the first author's name plus year of publication) that can be clicked to open the document in the browser; (ii) *Title:* When clicking on an item in the title column, the corresponding document gets selected in PaperViz (equivalent to selecting an entry in the tree view or node in the graph);

(iii) *Authors:*; (iv) *Add citation:* through a click action the document abbreviation is included in the text editor at the current or last known position of the cursor; (v) *Remove:* a click event removes the document from the bookmarks list.

Text editor

At the moment of conceptualizing PaperViz, the text editor was designed for drafting sections of a research paper. It later turned out that this component is more suitable for annotating bookmarked documents and sketch smaller portions of text. The citation function supports this purpose, as documents can be directly linked to text within the editor.

USER STUDY

We conducted a formative evaluation to determine if navigation, inspection, citation featores are well implemented and easy to use. Additionally, the study was intended to observe usage patterns. For instance, which of the three possible ways of navigating within a collection is preferred or how key phrases are placed within the visualization area.

Methodology

Participants had to perform four tasks with our tool. All tasks used the same sample collection, but each task had a different scope. After each task, participants had to answer questions assessing workload and level of difficulty. A post-study survey collected feedback about system features, usability and additional suggestions. The system recorded interaction logs for every action performed. An evaluator was present to assist participants in case of trouble and to take observational notes.

The data space consisted in sample hierarchically structured repository with 14 collections (including sub-collections) and 51 documents. The main 3 collections were "Visualization", "Clustering" and "Recommender Systems". All participants worked with the same data space. A session started with an introductory video explaining the system features and important visual encodings in detail. The next step consisted in a training task, where participants were encouraged to try as many features as possible and make all necessary questions. At this point the participant was ready to start the actual tasks. Each task description included a goal statement, necessary steps to be performed, useful hints and some questions to be answered at the end. They had to fulfill a total of 4 tasks:

Task 1: identify overall relevance of collections and documents (saturation). Participants also had to assess relevance of documents for a set of given key phrases by looking at bar charts and connecting edges.
Task 2: interpret the positions of circles within the visualization area to identify potentially relevant and irrelevant documents. In addition, this task required the use of navigation and filtering functionalities.
Task 3: use the tile bars to identify a document page where certain keyword appeared frequently. Moreover, they participants to refine the search query choosing phrases from the tag cloud or the MDA.
Task 4: formulate a “negative" query to identify documents and collections containing ‘clusters’ but weakly or not related to ‘tree’. In addition, participants had to bookmark documents and reference them in the editor.

After completing each task, participants answered 3 questions assessing perceived performance, effort and task difficulty. At the end of the study, they had to provide feedback about the system features and functionalities (16 questions) and fill a System Usability Scale (SUS) [5]. All form answeres were measured in a 7-point likert scale (-3 = strongly disagree, 3 = strongly agree). Finally, the examiner conducted an informal interview to validate observed behavior.

Results

A total of 16 participants took part in the evaluation, including novice and experienced researchers.

Workload and Task Difficulty

Boxplots in Figure 4 summarize user responses for perceived performance, effort and task difficulty. Overall, participants did not experience major problems fulfilling the 4 tasks, which is reflected on measurements of perceived performance. However, as participants were novice users of PaperViz, they had to learn how the different features work and to interpret the multiple encodings in the graph. Thus, effort and task difficulty were higher in the first two tasks. Moreover, task 1 was the first time participants had to work alone, hence their self assessment of performance is marginally lower than for tasks 2 and 3. Participants probably gained confidence thereafter. In turn, task 4 demanded the greatest effort. This was not entirely a surprise, since the goal required to formulate a partly ”negative" search query, i.e. identify documents strongly related to one key phrase but not related to another one. This is a rather uncommon search scenario when using a classic search engine. Therefore, some participants struggled to find the correct strategy to solve this task.

Overall Usability

For subjective assessment of system usability, participants filled a post-study *SUS* questionnaire. To keep consistency with the scoring scale throughout the whole survey, we used a 7-point likert scale instead of a 5-point one. User responses were multiplied by 1.66to obtain overall SUS scores in a range between 0 and 100. Thus, score s_i for question x_i was computed as $s_i = (x_i - 1) * 1.6$. PaperViz obtained a mean score of 89.3 ($SD = 7.38$), falling in the $98 - 100$ percentile range in the curved grading scale interpretation of SUS scores, which equals an **A+ grade**. We also calculated *Usable* (questions 1, 2, 3, 5, 6, 7 and 8) and *Learnable* (4 and 10) subscales [14], such that PaperViz scored 88.02 ($SD = 8.59$) and 94.79 ($SD = 5.99$), respectively, both equivalent to an A+ grade.

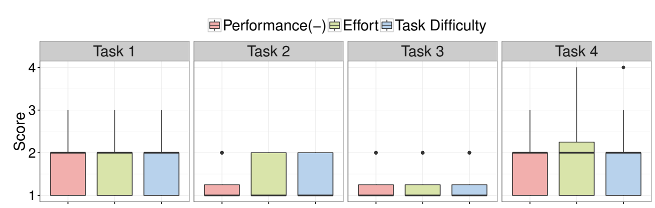


Figure 4: Workload measures in terms of perceived performance, effort and task difficulty across all tasks

Activity	Action	*mean* (*sd*)	*#users*
Navigation	via graph vis.	72.6 (64.2)	16
	via tree view	11.9 (7.9)	13
	via breadcrumbs	2.5 (1.7)	8
Key Phrase Discovery	added manually	11.6 (6.4)	16
	added from tag cloud	10.3 (4.9)	16
	added from MD area	1.2 (0.4)	6
Query Formulation	key phrase dropped	16.4 (9.9)	16
	key phrase moved	17.3 (12.5)	16
	key phrase removed	8.3 (7.6)	11
	gravity slider tuned	80.1 (53.9)	14
Details on demand	key phrase hovered	125.6 (101.9)	16
	collection hovered	89.8 (66.6)	16
	document hovered	114 (88.2)	16
	document clicked	9.7 (4.2)	16
Document Inspection	opened via tile bars	5.4 (3.7)	16
	opened via MD area	1 (0)	4
Resource Management	doc. bookmarked	5.8 (4.7)	16
	text editor used	5.6 (5)	16
	doc. cited	2.8 (1.7)	16

Table 1: Logged actions grouped by activity type.

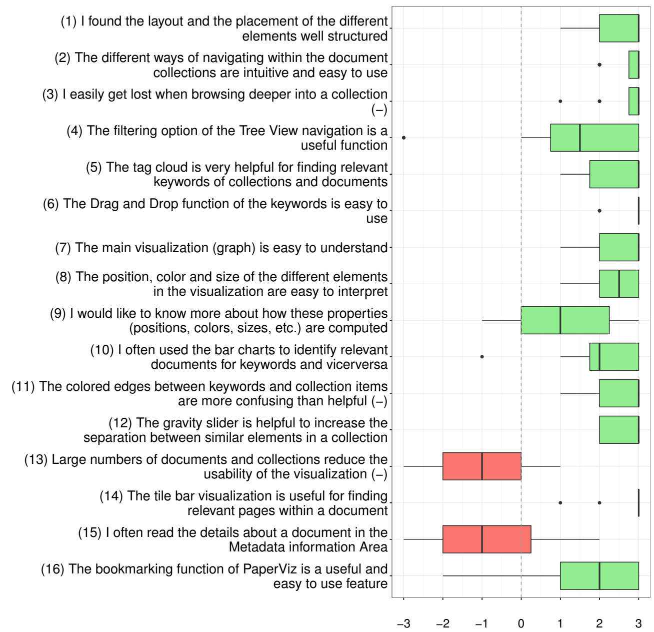


Figure 5: Subjective feedback for system features – questions marked with "(-)" are inverted

Usage and Feedback for System Features

To gain insight on the usage of the different components, we analyzed logged actions and grouped them by the kind of activity they involve. Descriptive statistics are presented in Table 1 (all 4 tasks). The results show a clear preference for the graph-based navigation, in contrast to the two other options, namely: via tree view or breadcrumbs. Only 8 out of 16 participants used the breadcrumbs to navigate to an upper level in the hierarchy, while the majority opted for the tree view despite being located farther from the visualization area. Perhaps users were not aware of breadcrumb-based navigation, in which case underlining the trails to emphasize their hyperlink character could improve this component.

Key phrase additions were performed manually or picked from tag cloud in similar proportions, whereas the MDA was the least frequent source for new key phrase. This highlights the usefulness of a topical overview for finding phrases of interest, in contrast to those found directly in documents. Query formulations involved manipulating tags within the visualization area. Participants performed over 16 tag drops on average, a similar number of occurrences for position shifts. Actions like moving key phrases around and the extensive use of the gravity slider imply a trial-and-error strategy to optimize the information conveyed by the layout.

Since most tasks required to find resources for a set of given terms, we expected to observe extensive mouse over events in the visualization area. Key phrases were the most frequently hovered nodes, followed by documents and collection nodes. Perhaps this indicates that putting the key phrase as reference node is the easiest way to assess relationship strength between a key phrase and a collection. Document clicks for intra-document details were less frequent and occurred mostly after filtering out irrelevant ones. In turn, users preferred to open PDF files directly on a specific page via the document's tile bars. Only 4 participants opened one document each by clicking 'open' in the MDA. Bookmarking and citation within the editor was only requested in task 4. Participants bookmarked and used the text editor 5 to 6 times on average and managed to cite roughly half of them.

We then analyzed the results of the responses about system features, summarized in Figure 5, along with our observations and comments gathered in the post-study interviews. The boxplots reveal two weaknesses at first glance (in red) and in further inquiries we spotted two other flaws to be improved.

Clutter (Q13): displaying a collection with dozens of nodes tended to hinder usability and readability of the graph visualization. Also, when many items have nearly identical relationship strengths to the chosen keywords, re-calcultaing their central points to avoid overlaps could cause that a circle ends up in a completely different position than originally planned, leading to a misinterpretation of the graph.
Few document reads (Q15): The MDA was not recognized as a valuable feature.This could be due to its placement in the GUI. The Gutenberg diagram defines the bottom-right portion of a page as terminal area, implying an inherent break in the reading or scanning process [18]. Also, low resolution screens force the user to scroll down.
Filtering (Q4): the only way to filter out irrelevant items is through the tree view, but first the user has to spot them in the graph. Thus, the user is forced the unintuitive behavior of removing a node in the graph via the tree view.
Bookmarking (Q16): In order to cite a document in the text editor, users have to perform three actions: select the document, click on "add to bookmarks" and finally click on "add citation" in the list of bookmarks. A more straightforward mechanism is needed, e.g. drag and drop.

Despite the identified weaknesses, the boxplots and SUS responses show that most functionalities received a positive rating. As the graph-based visualization is the core component of PaperViz, it should be noted that participants had no problems in understanding and interpreting it (Q7 and Q8). Final recommendations included support for drag-and-drop interactions with key phrases and LaTeX integration.

OVERVIEW AND FUTURE WORK

In this paper we presented a novel tool for tackling exploratory tasks in the context of scientific writing. We covered the design and interaction principles of the multiple coordinated views composing the UI and provided preliminary (mostly qualitative) results of a user study. Collected logged activity and user feedback allowed us to identify design pitfalls and pave the way for upcoming improvements. Notwithstanding, the outcomes are promising, considering that both experienced and novice users managed to complete tasks with different levels of difficulty and yielded a positive opinion afterwards. In the short term, we will focus on enhancing writing-support features, i.e. LaTeX integration and a full-sized editor. A more comprehensive user study is also due.

ACKNOWLEDGMENTS

This work is funded by the H2020 project AFEL (grant 687916). The Know-Center GmbH is funded by the Austrian COMET Program – managed by the Austrian Research Promotion Agency (FFG).

REFERENCES

1. Jae-wook Ahn and Peter Brusilovsky. 2013. Adaptive visualization for exploratory information retrieval. *Inf. Process. Manag.* 49, 5 (2013), 1139–1164. DOI: http://dx.doi.org/10.1016/j.ipm.2013.01.007
2. Keith Andrews, Wolfgang Kienreich, Vedran Sabol, Jutta Becker, Georg Droschl, Frank Kappe, Michael Granitzer, Peter Auer, and Klaus Tochtermann. 2002. The InfoSky Visual Explorer: Exploiting Hierarchical Structure and Document Similarities. *Inf Vis* 1, 3/4 (Dec. 2002), 166–181. DOI: http://dx.doi.org/10.1057/palgrave.ivs.9500023
3. Diane Blankenship. 2010. *Applied research and evaluation methods in recreation*. Human Kinetics.
4. David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.
5. John Brooke. 1996. SUS - A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7. DOI:http://dx.doi.org/10.1002/hbm.20701
6. Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. 2011. Apolo: Making Sense of Large Network Data by Combining Rich User Interaction and Machine Learning. In *Proc. CHI '11*. ACM, 167–176. DOI:http://dx.doi.org/10.1145/1978942.1978967
7. Chaomei Chen. 2006. *Information Visualization: Beyond the Horizon*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
8. Wenwen Dou, Xiaoyu Wang, Remco Chang, and William Ribarsky. 2011. Paralleltopics: A probabilistic approach to exploring document collections. In *Proc. IEEE VAST '11*. IEEE, 231–240.
9. Drahomira, Petr Herrmannova, and Knoth. 2012. Visual Search for Supporting Content Exploration in Large Document Collections. *D-Lib Magazine* 8, 7 (9 2012).
10. Jacob Eisenstein, Duen Horng Chau, Aniket Kittur, and Eric Xing. 2012. TopicViz: Interactive Topic Exploration in Document Collections *(CHI EA '12)*. ACM, 2177–2182. DOI: http://dx.doi.org/10.1145/2212776.2223772
11. Google. 2015. Word Trees. https://developers.google.com/chart/interactive/docs/gallery/wordtree#overview/. (12 2015). [Online; accessed 17-March-2016].
12. Brynjar Gretarsson, John Odonovan, Svetlin Bostandjiev, Tobias Höllerer, Arthur Asuncion, David Newman, and Padhraic Smyth. 2012. Topicnets: Visual analysis of large text corpora with topic modeling. *ACM TIST* 3, 2 (2012), 23.
13. Marti A. Hearst. 1995. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *Proc. CHI '95*. ACM Press/Addison-Wesley Publishing, 59–66. DOI: http://dx.doi.org/10.1145/223904.223912
14. James R. Lewis and Jeff Sauro. 2009. The factor structure of the system usability scale. *Lecture Notes in Computer Science* 5619 LNCS (2009), 94–103. DOI: http://dx.doi.org/10.1007/978-3-642-02806-9_12
15. Gary Marchionini. 2006. Exploratory Search: From Finding to Understanding. *Commun. ACM* 49, 4 (2006), 41–46. DOI:http://dx.doi.org/10.1145/1121949.1121979
16. David Newman, Timothy Baldwin, Lawrence Cavedon, Eric Huang, Sarvnaz Karimi, David Martinez, Falk Scholer, and Justin Zobel. 2010. Visualizing search results and document collections using topic maps. *Web Semant.* 8, 2 (2010), 169–175.
17. Kai A. Olsen, Robert R. Korfhage, Kenneth M. Sochats, Michael B. Spring, and James G. Williams. 1993. Visualization of a document collection: The vibe system. *Inf. Process. Manag.* 29, 1 (1993), 69 – 81. DOI: http://dx.doi.org/10.1016/0306-4573(93)90024-8
18. Heydon Pickering. 2016. *Inclusive Design Patterns*. Smashing Magazine. 312 pages.
19. Guy Shani and Noam Tractinsky. 2013. Displaying Relevance Scores for Search Results. In *Proc. ACM SIGIR '13*. ACM, 901–904. DOI: http://dx.doi.org/10.1145/2484028.2484112
20. Frank van Ham, Martin Wattenberg, and Fernanda B. Viegas. 2009. Mapping Text with Phrase Nets. *IEEE TVCG* 15, 6 (Nov. 2009), 1169–1176. DOI: http://dx.doi.org/10.1109/TVCG.2009.165