Denis Munter, BSc

Image-Based Visual Document Retrieval and Exploration

MASTER's THESIS

to achieve the university degree of
Master of Science

Master's degree programme:
Computer Science

submitted to
Graz University of Technology

Supervisor

Tobias Schreck, Univ.-Prof. Dipl.-Volksw. Dr.rer.nat. M.Sc.
Institute of ComputerGraphics and KnowledgeVisualisation, TU Graz

Graz, November 2020

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date, Signature

Acknowledgment

First of all I would like to thank all those who supported and motivated me during the preparation of this master thesis.

Special thanks go to my supervisor Prof. Dr. Tobias Schreck and my tutors Lin Shao and Hendrik Lücke-Tieke not only for proofreading my master thesis, but especially for their helpful suggestions and constructive criticism during the preparation of this thesis.

I would also like to thank my fellow students, who supported me with their interest and helpfulness but most of all for the numerous interesting debates and ideas which contributed significantly to the fact that this master thesis not only helped me to grow beyond myself but also was an instructive and interesting journey for me.

And my biggest thanks go to my family for all the support you have shown me through years of distance learning. And of course my girlfriend Sharon, thanks for all your support, your helping hand and your advises whenever I needed it, without which I would have stopped these studies a long time ago, You have been amazing!

Abstract

Nowadays document retrieval systems are used to obtain information about document collections. However, often only their textual data is included and not additional data, such as the images they contain. For this reason the motivation for this master thesis was to create a tool that supports not only the analysis of text data but also images and the combination of both. In order to support image and text analysis, specially developed methods were selected and integrated into the software. Examples are the creation of image clusters and topics, visualization of documents using document cards and an image browser to analyze the images of the documents. Thus, the developed tool not only guarantees the user a first interesting overview of any document collection, but also provides several useful visualization methods. Furthermore, relations between topics and image clusters can be investigated and, if necessary, manually created topics can be used to find better relations based on predefined text or image queries.

Contents

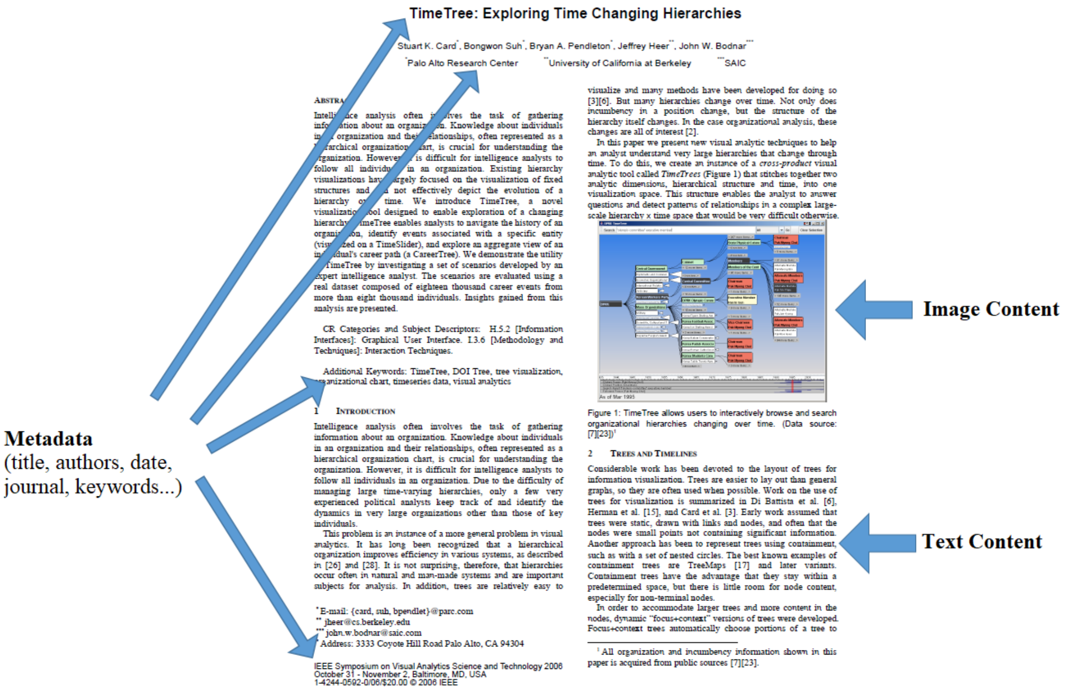
Chapter 1

Introduction

Many researchers are interested in content related works that address and support their particular areas and research topics. For instance, for the visualization community the visual results of a paper are the most important ones. But if we look for some slightly related approaches (e.g. part of a new developed technique, alternative representations, further developed algorithms or even new found insights which don't address the main content or topic of a paper) we run into a common problem which comes with basic textual search that is: Smaller contributions, which are not indexed or strongly stressed in the paper, can be hard to find and normally that document retrieval systems are overall text-based and don't include image similarity measures which means the retrieval relies only on text-based queries with textual annotations like title, author, date, journal, keywords or other meta-data. However, in many cases also visual content is important in documents, not only to provide a solution for the problem of finding additional information with the support of images but also to find papers based on similar example visualizations, which is difficult by relying on keyword search alone. [SGG*18]

There exist many techniques for content-based image retrieval [SWS*00] but how do we visualize the gathered information or how do we generate a proper uniform view for the received results? In general an information retrieval system can be defined as a system which describes the dynamically prioritization of search request results. The goal is to develop an information retrieval system where the results can always be further investigated to achieve a dynamic overview about the results. One suggestion would be to express the corresponding information from documents and the individual gained results as a graph where, e.g., the documents are ranked with respect to a query, upon relationships among documents, relationships among topics, and relationships between query terms and documents [TS06]. Another solution could be a graphical simplification of a document's overview. For example in [SOR*09a], a compact visualization of documents, demonstrated for the IEEE InfoVis publications of a complete year, is presented like cards in a top trumps game where the document's key semantic is illustrated, not only using their textual information but a mixture of images and important key terms. This provides not only the above mentioned combination of image and text information but also a suitable less space consuming view for a document's main content which is also crucial to provide if we access large document collections. We can see it is not only a question how do we extract information, it is even more important to develop proper visualization techniques for the gathered results and how do we present them in a way so it can be easily understood and be further investigated by the underlying users.

In summary we can say there exist many approaches for document collection analysis based on text and metadata. On the other hand less support exists for analysis of image content in document collections and even lesser for the combined image and text/metadata bases analysis. Therefore the goal of this thesis is to develop and apply a visual exploration system to analyze document collections from the images and text/metadata they contain and try to find relations between images and the text data to provide more in-depth details about the overall document collection.



TimeTree: Exploring Time Changing Hierarchies

Stuart K. Card*, Bongwon Suh*, Bryan A. Pendleton*, Jeffrey Heer**, John W. Bodnar***

*Palo Alto Research Center **University of California at Berkeley ***SAIC

ABSTRACT

Intelligence analysis often involves the task of gathering information about an organization. Knowledge about individuals in an organization and their relationships, often represented as a hierarchical organization chart, is crucial for understanding the organization. However, it is difficult for intelligence analysts to follow all individuals in an organization. Existing hierarchy visualizations have largely focused on the visualization of fixed structures and can not effectively depict the evolution of a hierarchy over time. We introduce TimeTree, a novel visualization tool designed to enable exploration of a changing hierarchy. TimeTree enables analysts to navigate the history of an organization, identify events associated with a specific entity (visualized on a TimeSlider), and explore an aggregate view of an individual's career path (a CareerTree). We demonstrate the utility of TimeTree by investigating a set of scenarios developed by an expert intelligence analyst. The scenarios are evaluated using a real dataset composed of eighteen thousand career events from more than eight thousand individuals. Insights gained from this analysis are presented.

CR Categories and Subject Descriptors: H.5.2 [Information Interfaces]: Graphical User Interface. I.3.6 [Methodology and Techniques]: Interaction Techniques.

Additional Keywords: TimeTree, DOI Tree, tree visualization, organizational chart, timeseries data, visual analytics

1 INTRODUCTION

Intelligence analysis often involves the task of gathering information about an organization. Knowledge about individuals in an organization and their relationships, often represented as a hierarchical organization chart, is crucial for understanding the organization. However, it is difficult for intelligence analysts to follow all individuals in an organization. Due to the difficulty of managing large time-varying hierarchies, only a few very experienced political analysts keep track of and identify the dynamics in very large organizations other than those of key individuals.

This problem is an instance of a more general problem in visual analytics. It has long been recognized that a hierarchical organization improves efficiency in various systems, as described in [26] and [28]. It is not surprising, therefore, that hierarchies occur often in natural and man-made systems and are important subjects for analysis. In addition, trees are relatively easy to visualize and many methods have been developed for doing so [3][6]. But many hierarchies change over time. Not only does incumbency in a position change, but the structure of the hierarchy itself changes. In the case organizational analysis, these changes are all of interest [2].

In this paper we present new visual analytic techniques to help an analyst understand very large hierarchies that change through time. To do this, we create an instance of a *cross-product* visual analytic tool called *TimeTrees* (Figure 1) that stitches together two analytic dimensions, hierarchical structure and time, into one visualization space. This structure enables the analyst to answer questions and detect patterns of relationships in a complex large-scale hierarchy x time space that would be very difficult otherwise.

Figure 1: TimeTree allows users to interactively browse and search organizational hierarchies changing over time. (Data source: [7][23])[1]

2 TREES AND TIMELINES

Considerable work has been devoted to the layout of trees for information visualization. Trees are easier to lay out than general graphs, so they are often used when possible. Work on the use of trees for visualization is summarized in Di Battista et al. [6], Herman et al. [15], and Card et al. [3]. Early work assumed that trees were static, drawn with links and nodes, and often that the nodes were small points not containing significant information. Another approach has been to represent trees using containment, such as with a set of nested circles. The best known examples of containment trees are TreeMaps [17] and later variants. Containment trees have the advantage that they stay within a predetermined space, but there is little room for node content, especially for non-terminal nodes.

In order to accommodate larger trees and more content in the nodes, dynamic "focus+context" versions of trees were developed. Focus+context trees automatically choose portions of a tree to

*E-mail: {card, suh, bpendlet}@parc.com
** jheer@cs.berkeley.edu
*** john.w.bodnar@saic.com
*Address: 3333 Coyote Hill Road Palo Alto, CA 94304

IEEE Symposium on Visual Analytics Science and Technology 2006
October 31 - November 2, Baltimore, MD, USA
1-4244-0592-0/06/$20.00 © 2006 IEEE

[1] All organization and incumbency information shown in this paper is acquired from public sources [7][23].

Figure 1.1: Different data types inside a document

The difficulty hereby is often not only the extraction of the relevant information by its own but also to create and more important to present meaningful results using various visualization tools that can be selected and even combined by the user. Hence the main focus of the master thesis lies on the different visualization and analysis techniques. Important questions will be: Can we find any kind of meaningful image clusters and topics that exist in certain document collections? How can we visualize them and how can we find relations between those topics and image clusters and does that help us to find new topic related documents? How can we visually compare metadata and text properties of documents and does the clustering allow to classify new papers with respect to existing topics.

This paper is organised as follows. Section 2 discusses the related work on information retrieval systems and some of their visualization approaches. Section 3 describes the concept of our own exploration system followed by the detailed description of the overall implementation in Section 4. Section 5 describes our use cases, the corresponding findings and evaluations. After that, in section 6 we give a short discussion about the limitations of our developed exploration system. The master thesis concludes in Section 7.

Chapter 2

Related Work

2.1 Information Retrieval Systems

2.1.1 Definition

Information retrieval (IR) is defined as the organisation, storage and maintenance of information from document collections [KM02]. The main objective of an information retrieval system is to find the desired information and its relevant documents. When we talk about information we usually speak about text but also video, audio and other multimedia. The basic workflow is the formulation of a request in the form of a query and then the IR system will respond by retrieving the relevant output. Information retrieval systems are based, either directly or indirectly, on models of the retrieval process [Cro06]. These models are very important to help us designing and implementing an efficient information system. Furthermore they predict and explain what a user will find in relevance to a given query. In theory an IR model consists of a model for documents, a model for queries and a matching function that compares queries to documents. In terms of content we can describe them as a representation of documents and queries as well as a ranking method (similarity function which orders the documents with respect to a query). In [MRBK17], these models are classified as follows:

1. Classical models (e.g. boolean and vector space model)
2. Probabilistic models (e.g. BM25 or language model)
3. Combining evidence models (e.g. inference network and language to rank model)

Boolean models provide exact matching by using logical operators (and, or, not) and don't involve some sort of ranking for documents [GD09]. The retrieval is based on whether or not the documents contain the query terms. On the other hand in vector space models the text documents are represented as vectors that will be used to calculate the similarity between documents and queries through the euclidean distance and cosine similarity. However, in contrast to the previous model a ranking is supported. [MRBK17]

Probabilistic models use conditional probability and require two conditions: the relevant document, which is obtained by computing the probability of containing the specific document terms, and the long queries that distinguish between the presence and the absence of terms inside documents [GD09]. The advantage of these models is that they don't need an additional term weighting algorithm, but there is no accurate estimate for the first run probabilities and since the conditional probability is used, the terms are assumed to be mutually independent. [MRBK17]

Combining evidence models aim for the combination of multiple sources of evidence for improving the effectiveness of the overall information retrieval process. [TL04]

2.1.2 Design features

Inverted Index

In case of dealing with very large document collections the primary data structure of most IR systems is the inverted index, a database index storing a mapping of content [Knu97]. To map from documents to content, every term will be indexed with all documents containing it including the corresponding term frequencies or other document specific data. The overall goal is to optimize the speed of a text query (find documents containing a specific word or text).

Stop Word Elimination

Words that are used very often in natural language have less semantic weights. That's the reason why these terms have to be excluded from the entire vocabulary set before the IR system actually tries to index them. Creating a list of that words, also called stop words, can significantly reduce the size of an inverted index. However, stop word elimination does not always bring better results because it might eliminate terms that are useful for searching. A simple example would be the elimination of B in the term constellation "$Vitamin\ B$" that would change the actual semantic meaning. Therefore the general trend in IR systems over time has been from standard use of quite large stop lists (200 to 300 terms) to very small stop lists (7 to 12 terms) to no stop list whatsoever. [Knu97]

Stemming and Lemmatization

The stemming or lemmatization method are key steps in english document processing and can be named as normalization. However they differ from each other in algorithm and processing result. The stemming approach reduce a word to its stem or root form by looking for prefixes or suffixes and removes them. Lemmatization on the other hand aims to remove inflectional endings only and return the base from of a word (called lemma). If we look at the result, the stemming algorithm returns a token that may have no meaning at all whereas the lemma transformed by the lemmatization is a actual real word from the vocabulary. In terms of usage, stemming is mainly used in information retrieval while lemmatization most common usage lies in machine translation where accurate word tokens are required. [HSWL12]

Relevance Feedback

Relevance in information retrieval defines how the retrieved information meets the requirements of the users. Hence it plays a very important role in IR systems because it helps to improve the performance and accuracy of retrieval systems. Since the user him/her-self is responsible for the formulation of the queries, it is quite obvious that the output of any IR system is depending on the user. A well structured or formatted query will probably produce more accurate results than badly formatted ones. Therefore the primary goal of relevance feedback is not only to take out the initially returned queries and use the outputs to gather user information but also use user feedback in direct constellation to the results (do the results show relevancy or not) and present that information to the IR system. The system then uses this information in two ways: quantitative approach (retrieve more relevant documents) and qualitative approach (retrieve similar documents that are relevant). As shown in figure 2.1, the process itself can be seen as a cycle of activities. First the user submits a query, the IR system performs the query and returns the results. Then the user gives feedback according to the results and after that the relevant results will be reformulated based on that given feedback. [Ham17]

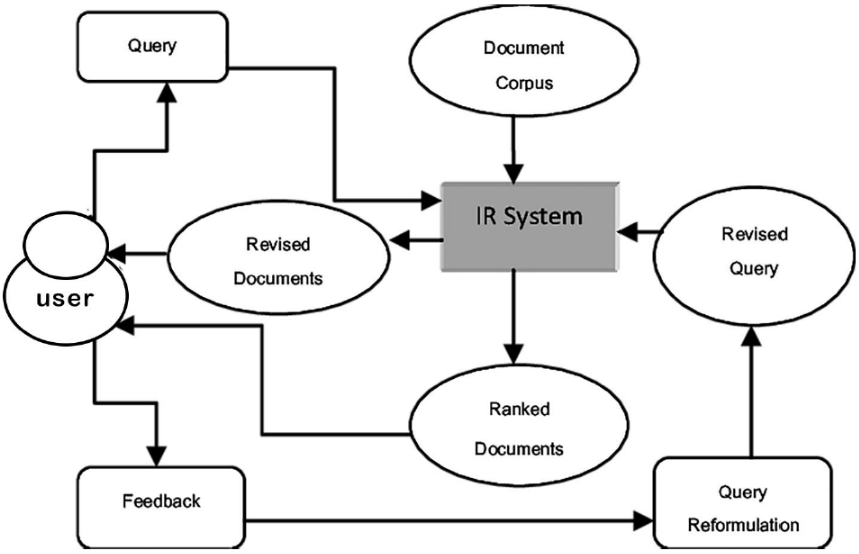


Figure 2.1: Basic architecture of feedback [Ham17]

According to Anmol Hamid [Ham17], there exist three different types of feedback:

1. **Explicit feedback**
   The user indicates the relevance of the document that are retrieved by a query. The binary relevance system states if a document is relevant or not and the graded relevance system uses some kind of description (words or scaling through numbers).

2. **Implicit feedback**
   This feedback is indicated by noting users behavior (e.g. what kind of document do they view or the time spent on viewing a document).

3. **Pseudo feedback**
   It follows the principle: ”What comes first is more relevant than what follows”. It automates the manual part of relevance feedback, so that the user gets improved retrieval performance without an extended interaction [MRS08]. The system returns relevant results and assumes that the top k documents are the most relevant ones. Then the top 10-25 terms from these documents are selected by using tf-idf weights. After that the query is expanded by adding those terms. Last but not least the returned documents are getting matched for the new query and the final output is returned which shows that most of the relevant documents are returned.

2.1.3 Evaluation

We have already described many alternatives in designing an information retrieval system but how do we know which of these techniques are effective? In order to answer that question we have to evaluate the IR system by measuring its effectiveness. The two most frequent and basic measures for information retrieval effectiveness are precision and recall [MRS08]. Precision is the fraction of retrieved documents that are relevant and Recall is the fraction of relevant documents that are retrieved.

$$Precision = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} \qquad Recall = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})}$$

To make both this notions clearer, Manning [MRS08] stated the following contingency table:

	Relevant	Nonrelevant
Retrieved	true positives (t_p)	false positives (f_p)
Not retrieved	false negatives (f_n)	true negatives (t_n)

With the above stated definitions we can formulate Precision P and Recall R as:

$$P = \frac{t_p}{t_p + f_p} \qquad R = \frac{t_p}{t_p + f_n}$$

Another possibility to evaluate an information retrieval system would be to determine its accuracy. According to Manning, an IR system can be seen as a two-class classifier which classifies the retrieved documents to either relevant or nonrelevant. In terms of his contingency table, he defines the accuracy as follows:

$$Accuracy = \frac{t_p + t_n}{t_p + f_p + f_n + t_n}$$

But there is a good reason why accuracy is not an appropriate alternative measure to precision and recall. If we are only looking onto the performance, a system which maximizes the accuracy can perform well by simply labeling all query related document results as not relevant. Even if the system works quite good and the results are categorised as relevant it will almost always lead to a high rate of false positives. Since the user is not interested in finding non-relevant documents the measure of precision and recall is the more sensible option because it concentrates the evaluation onto the return of true positives by stating the percentage of relevant documents that have been found in contrast to the returned false positives. In order to actually use the mentioned evaluation approach, certain test collections are needed. Examples for such collections would be the *Text Retrieval Conference* (TREC) with over 528,000 newswire and Foreign Broadcast Information Service articles, the *Cross Language Evaluation Forum* (CLEF) which concentrates on european languages and cross-language information retrieval or the widely used text classification collection called *20 Newsgroups* collected by Ken Lang. [MRS08]

2.2 Visualization

2.2.1 Definition

Visualization is the process of transforming information into a graphic representation to support data analysis and information exploration. Hence it enables researchers to explore the content of documents from different aspects and different levels of details. The whole observation may also provide them unexpected insights from their simulations or computations. In general it requires certain methods or algorithms to convert information into a meaningful form so that users can interpret them from a whole new perspective. In that sense the visualization is a kind of communication process between information and the user which offers a visual data analysis that outperforms the standard numerical or statistical methods because of the new obtained data contexts and relationships. There exist many document visualization techniques which can differ in visualizing unstructured data (text) and structured data. We will discuss some examples for such visualization techniques which are also relevant for this master thesis. We start by looking at the text visualizations that don't represent the text directly, e.g. word count or word sequences. Examples would be Phrase Nets or advanced tag cloud approaches like ConcentriCloud and Affective Word Clouds. On the other hand we have the visualization of image data that makes use of the images in their original visible format. Examples would be the Slice Histogram or the Growing Entourage Plot. In the end we discuss a method that combines the visualizations of text and images: the Document Cards. [Zha08, Cro16, SOR*09b]

2.2.2 Tag Clouds

A tag cloud (also called word cloud) is a method of information visualisation that provides a first impression of text documents by displaying a list of keywords depending on their different weightings. In principle, the font size of a keyword in a word cloud is determined by its frequency. Tag clouds are widely used for non-analytic purposes. However, normal word cloud visualizations only provide limited support in comparing the words and word frequencies of different text documents. For this purpose ConcentriCloud was proposed to systematically merge and display the words from several text documents. Basically it is a combination of several word clouds from different documents that offers comparable views, avoidance of redundancies and an identification of commonalities and differences between the involved documents. The described features will be achieved by its composition principle where the world clouds are arranged as concentric circles, sketched in figure 2.2. [LHB*15]

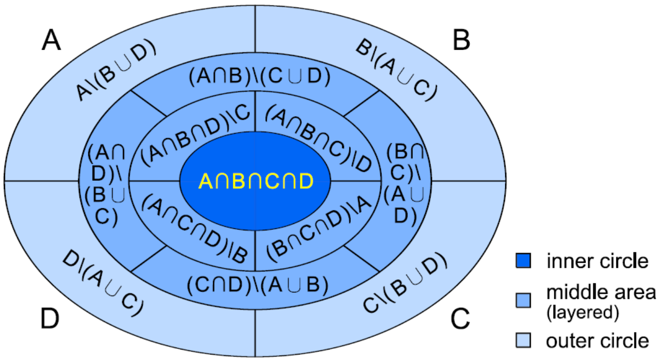


Figure 2.2: Composition of ConcentriCloud (letters A to D represent words of four documents) [LHB*15]

As shown in figure 2.3 they created a ConcentriCloud example by visualizing the frequent terms of all seven "Harry Potter" novels. The outermost circle represents the individual novels (HP1 to HP7) which are also visually separated by lines.

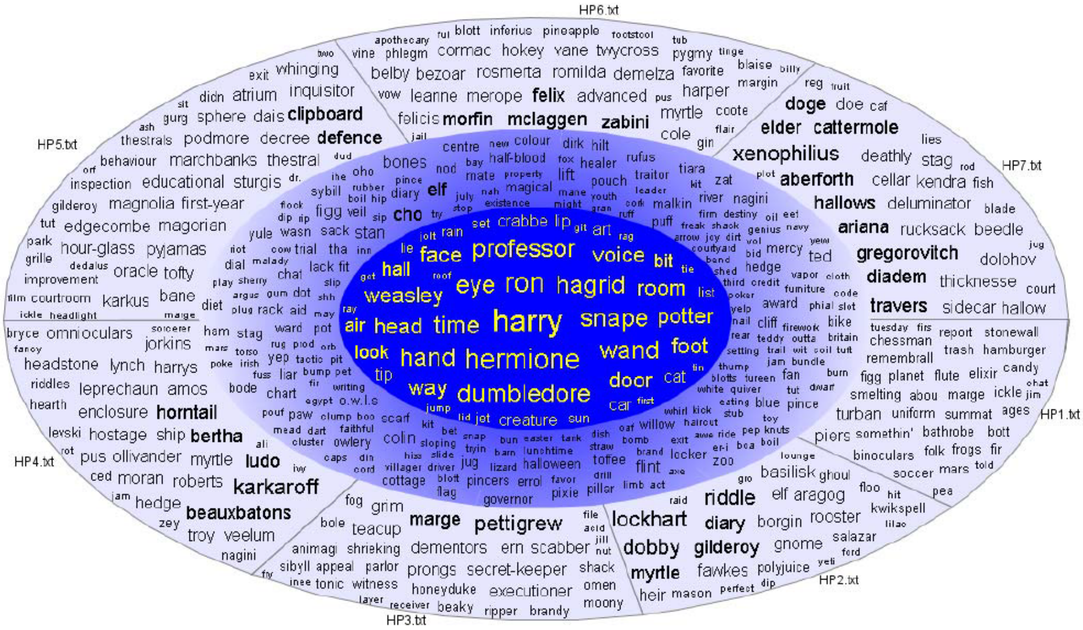


Figure 2.3: ConcentriCloud example of the Harry Potter novels [LHB*15]

As practical as the ConcentriCloud is, it has a certain limitation as well. Sure it can visualise the words from an arbitrary number of documents, but it would usually not make much sense to visualize words of more than a dozen of documents because this would become too demanding for the viewer. [LHB*15]

Tag clouds can also be used to gain specific insights in context of semantic meaning by for example placing semantically similar words closer to each other. Even further goes a study of Tugba Kulahcioglu and Gerard de Melo [KdM19] in which the possibility of generating tag clouds with a stronger emotional impact is considered. They tried to achieve that impact by introducing fonts and color palettes as powerful paralinguistic signals. They created a tool that takes a text and the desired affective preferences as an input and then the tag cloud is generated by using a self implemented specific font selection and semantic color palettes. The color palettes were taken by Lyn Bartram who was of the opinion that different color properties contribute to different affective interpretations in information visualization [BPS17]. An example of this procedure is shown in figure 2.4 where a colorful tag cloud of the United Nations based on paralinguistic signals is shown.

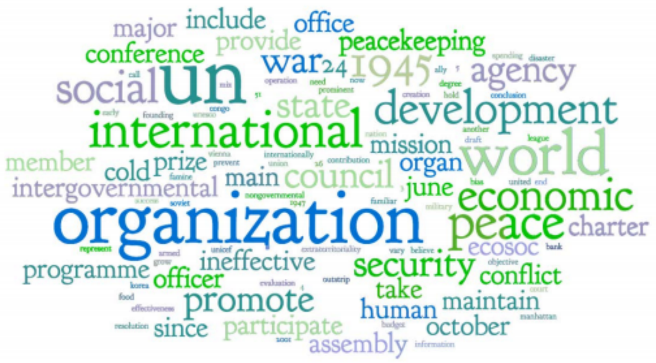


Figure 2.4: Affective word cloud [KdM19]

In summary we can say that tag clouds are useful to provide a summarising overview over a topic given a certain text or even multiple documents and it is also revealed that both fonts and colors can be used as additional dimensions for tag clouds to intentionally encode affect. Furthermore any visualization that makes use of fonts or color palettes should be considered of using the affective nature of paralinguistic signals.

2.2.3 Phrase Net

A further technique for generating visual overviews of unstructured text is Phrase Net which displays words and their corresponding relations between each other as a linked graph of nodes. To create such a network the nodes are defined to be a subset of words occurring in the text and the edges represent certain relations between these words which can be defined by the user at either a syntactic or lexical level. To find the connections between words two methods were investigated: Syntactic linking and orthographic pattern matching. Syntactic linking uses patterns based on the syntactic structure of sentences. The sentences are split up into words and then linked together by several types of relations chosen by the user. This allows the user to spotlight certain types of relationships and in addition avoids obvious dependency problems (in a sentence like “the keys were found,” we will know that “keys” and “found” are related). But this procedure was very time consuming. The more rapid exploration was the simple text based pattern matching using regular expressions that can be done at interactive speeds. After parsing the text, they got a directed graph representing the order the words occurred with a weight depending on the quantity a matching pattern was found. This graph was then filtered to reduce its size by removing the most common words (stopwords). Figure 2.5 shows an example of such a network. As we can see, the visual representation of the results includes not one but several clusters composed with either a single or a collection of topologically equivalent nodes. [HWV09]

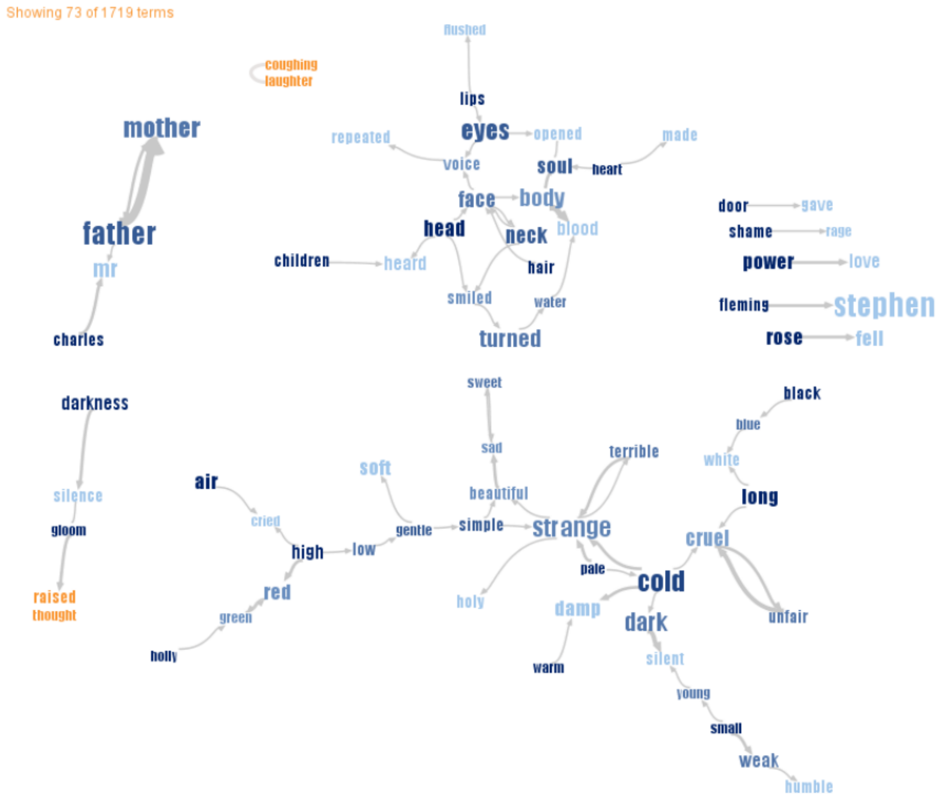


Figure 2.5: Phrase Net applied to James Joyce’s Portrait of the Artist as a Young Man [HWV09]

2.2.4 Slice Histogram

General tools like image histograms have a great variation in their application. A basic form of a image histogram shows the distribution of a single metric (e.g. brightness, time, geolocation, etc.) and uses images as plot elements. Within the plot, every image has its own place and the axes serve as organization or otherwise sorting of that data points to reveal patterns. In other words, it is a collection of data points plotted along two axes that reveals the relationship between the variables mapped to the axes. Hence we can assume that each histogram can be sorted as many times as we like, as shown in figure 2.6. But the problem is that images typically could contain a lot of different colors. For this reason the question arises: how do you sort images by color? Do you take the mean hue of the whole image or do you look at the color dimensions? To find a proper solution, Damon Crockett assumed that images have a very low standard deviation of their color properties. This means that the more uniformly-colored the images are, the easier it is to sort them by color. But it is obvious that it is not possible to simply enforce uniformity in image data because in general it is quite difficult to see the basic visual properties like hue, saturation and brightness. Based on the fact that images capture scenes and scenes have different parts, he came up with the idea to just plot slice a image instead of just displaying the whole image. In other words, in slice histogram images will be sliced into visually homogeneous parts and that parts will be used as the histogram elements. But it turned out that this procedure is very computational expensive, so the goal was to find a better color visibility without loosing computational speed. [SHD, Cro16]

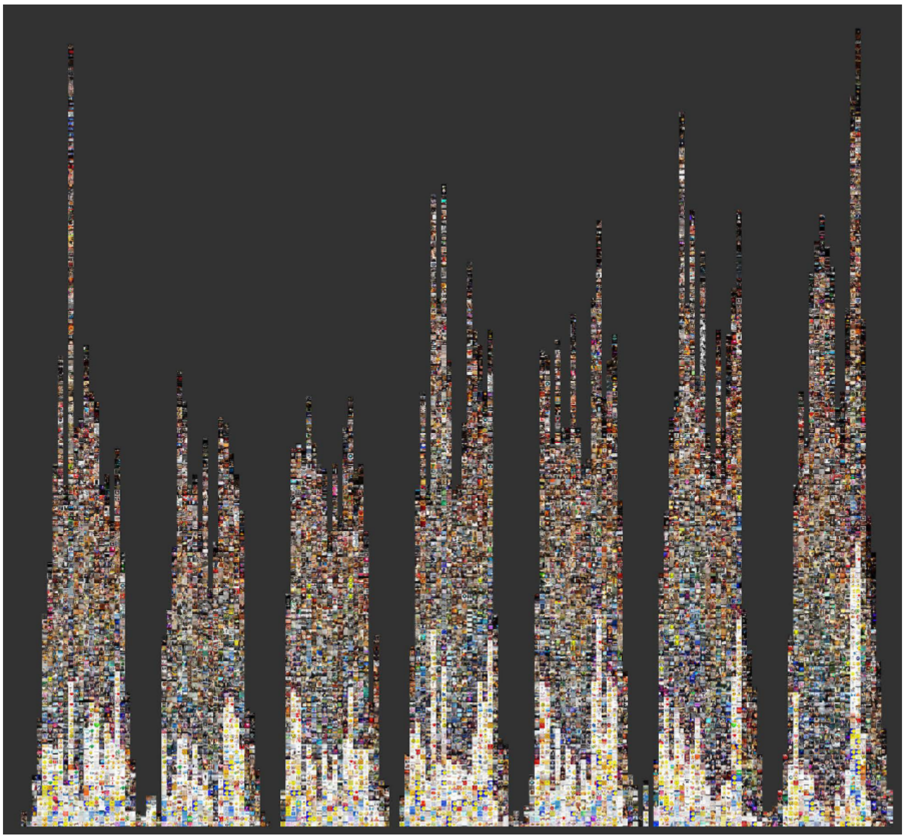


Figure 2.6: Multiple sorted image histograms (sorted vertically by both brightness and image hue) [Cro16]

Therefore each image is sliced into a number of equal-sized parts from which simultaneously color properties are extracted. Then these image parts are plotted as an image histogram. An example for this approach is shown in figure 2.7. The left part shows the histogram and the right part is a close-up. The overall number of parts depend on the kind of images that are used. Damon Crockett set an average standard deviation of hue (~ 0.1) to find the minimum number of parts. This ensures a smooth and consistent presentation of color and as much object content as possible for every image data. In general it is essential that the plot elements have a low standard deviation of visual properties and the integration of a sorting method that groups together similar elements. [Cro16]

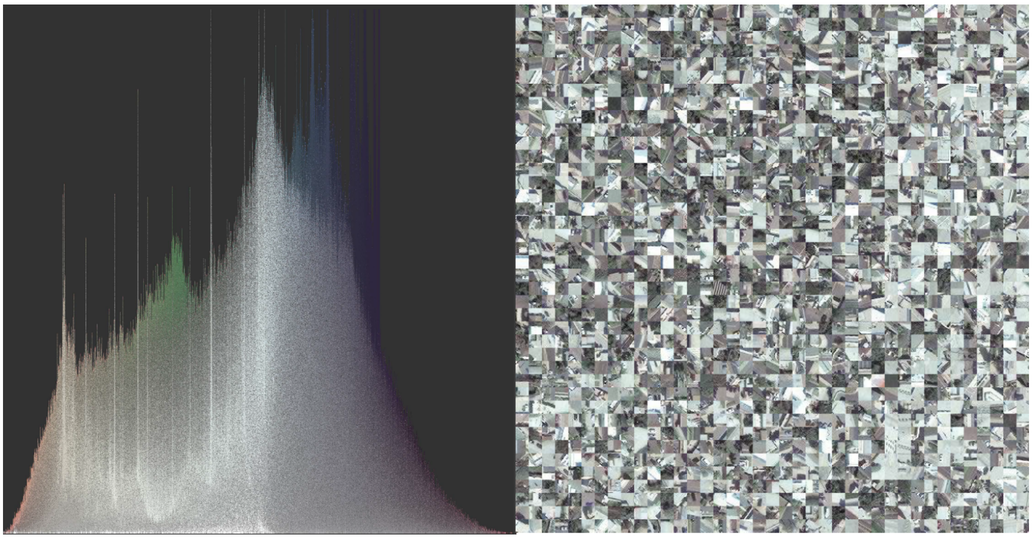


Figure 2.7: One million slices of a satellite image, arranged as a hue histogram sorted vertically by saturation [Cro16]

2.2.5 Growing Entourage Plot

In information visualization a lot of research is spent on the problem of how to present high-dimensional data on two-dimensional canvases. According to Damon Crockett [Cro16] there are at least three broad categories of solution:

1. **Preserve all features (visualizing everything)**
   Such visualizations can be difficult to design and to read because additional choices about sorting can make big differences in readability.

2. **Preserve some by selection**
   Here we select some subset of features and use them for sorting, e.g. by only brightness and hue. This is powerful and useful but on the other hand it makes very complex sorts of similarity relations between images invisible.

3. **Preserve some by redefinition**
   Reduces the dimensionality of the data by defining a new and compact feature space in the way that it can reveal very complex sorts of similarity between images.

But there is also a completely different approach which deals with dimensionality: Clustering. But the first problem already occurs in the visualization of the clusters. Since the human visual system can analyze a limited number of spatial dimensions, it is difficult to display relational image data because they already carry information only by their spatial positions. The growing entourage plot provides a solution by visualizing high-dimensional image clusters using only two dimensions. In other words it projects cluster centroids into the two-dimensional space and builds clusters around them by turn-taking and semantic priority. [Cro16]

At the beginning they start with high-dimensional image data and use the k-means algorithm to find K clusters. Then each centroid will be projected to the two-dimensional space by using a dimensionality reduction algorithm like PCA or t-SNE. Afterwards these coordinates will be binned to a grid. The result is a complex similarity relations among cluster centroids. The last remaing step is to build the clusters on the grid at the two-dimensional centroid locations. For this purpose they used the Euclidean Distance [FPR09] to assign each image to a cluster and simultaneously rank them by the distance difference to the cluster centroids. As shown in figure 2.8, the result is a scatter plot with image centroids (empty spaces) surrounded by their high ranked cluster members. The remaining low ranked members are scattered in nearby territories. [Cro16]

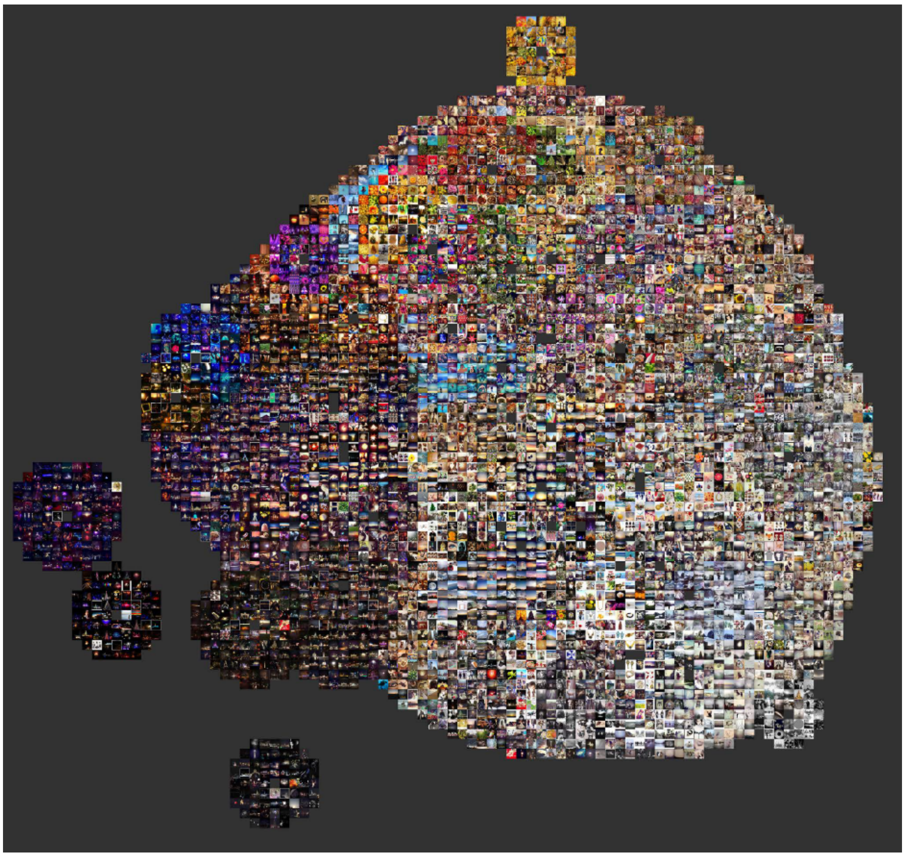


Figure 2.8: Example of a Growing Entourage Plot [Cro16]

2.2.6 Document Cards

Today, search engines usually display the title of a document along with a small context of the query terms. This will make a user only need to read parts of the text and in addition keeps the focus on the relevant parts of the documents, which allows the user to efficiently distinguish between relevant and non-relevant documents. Nevertheless, this representation is efficient for browsing through simple search results but not to give a compact overview of a single document or even document collections. Therefore a compact visual representation of documents was developed, the Document Cards (DC). They use important keywords and important images to maintain the informative value of texts and combine it with the descriptive nature of images in one view. Figure 2.9 shows the pipeline for creating Document Cards, which basically consists the following steps: extraction of the text data and the additional determination of key terms; image extraction; generation of the corresponding document cards by creating a layout involving both, image and text data. [GZL*14, SOR*09b]

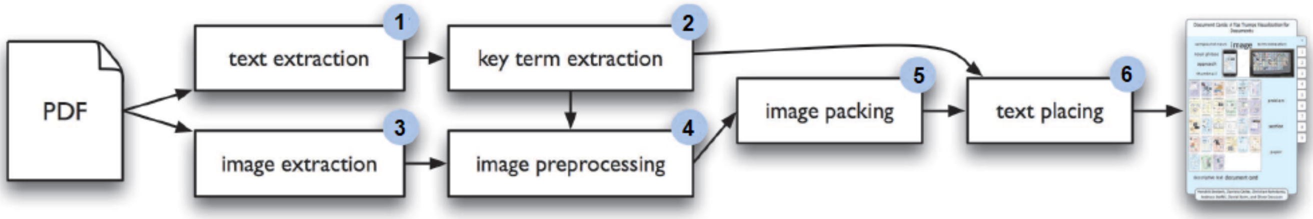


Figure 2.9: The Document Card pipeline [SOR*09b]

The design of the document cards plays the most important role because it is the direct intersection point between the information retrieval system and the user. Document Cards are fixed size thumbnails that are selfexplanatory, hence it requires a decision of what can be preserved and what has to be excluded. Erol et al. [EBJ06] evaluated the most important parts of a document, which are: title, images, and the abstract. From this the document cards were designed to include a filtered collection of images and the important keywords as an approximation for the document content. The positioning of the images is done iteratively on the Document Card according to the coordinates that are given by the packing algorithm defined in [SOR*09b]. At the same time the free areas left on the cards will be collected where later the keyterms will be placed. Additionally the document card is enriched with the document title and the documents author names. Furthermore a page number list will be added at the right side of each card to show the overall size and to navigate through the document. To make it easier for the user to actually distinguish between different document cards, the most frequent color value of the corresponding document images will be evaluated and the background is coloured accordingly (examples in figure 2.10). [SOR*09b]

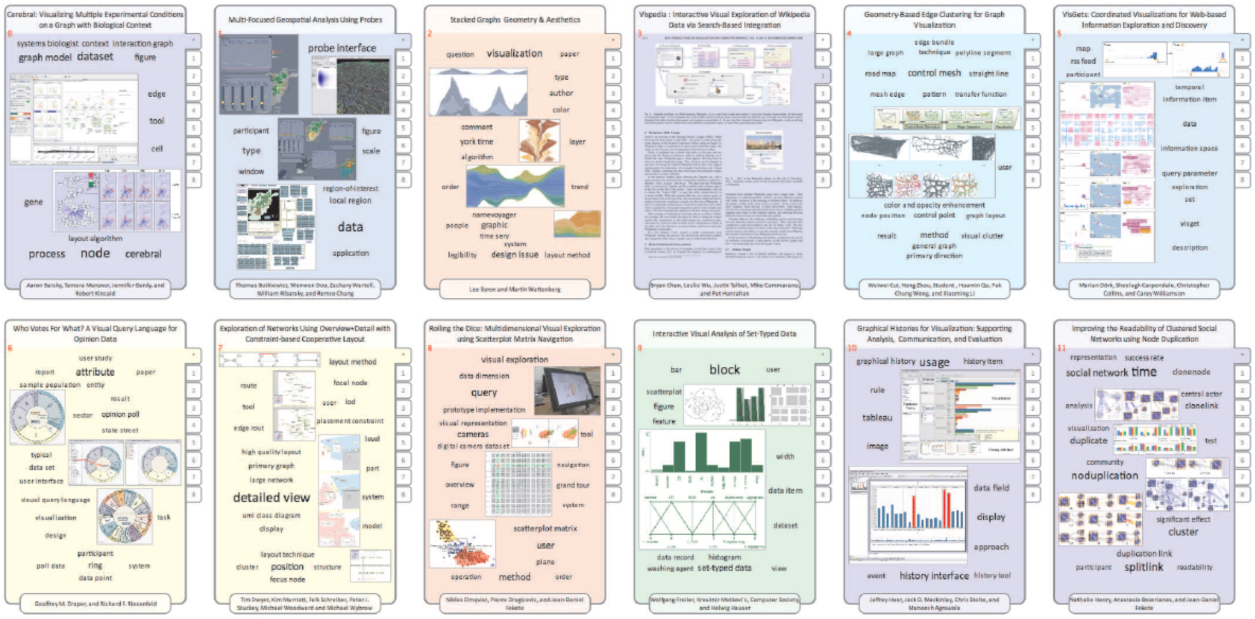

Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context
Multi-Focused Geospatial Analysis Using Probes
Stacked Graphs Geometry & Aesthetics
Vispedia: Interactive Visual Exploration of Wikipedia Data via Search-Based Integration
Geometry-Based Edge Clustering for Graph Visualization
VisSets: Coordinated Visualizations for Web-based Information Exploration and Discovery
Who Votes For What? A Visual Query Language for Opinion Data
Exploration of Networks Using Overview+Detail with Constraint-based Cooperative Layout
Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation
Interactive Visual Analysis of Set-Typed Data
Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation
Improving the Readability of Clustered Social Networks using Node Duplication

Figure 2.10: Document Card Examples [SOR*09b]

Chapter 3

Concept

3.1 Extraction of document content

Before we deal with the content of a document collection, we must first have the possibility to extract the necessary data from the documents. Therefore the exploration system should provide an extraction feature which, after specifying a source folder, loads all documents into the system and reads the required data from the papers. Furthermore, we should also be able to export/import already extracted documents and their corresponding meta-data. This should serve as a workaround in order to avoid having to extract old document collections that have already been sufficiently examined.

3.2 Image- and Document Browser

When we work with a document collection for the first time, we do not know exactly what the collection is all about. We are facing the following questions: What are the general topics of the documents, which images are involved, when were the documents published, what authors do they have and many others. The Image- and Document Browser should answer those questions by giving a first overview about the document collection and showing us all documents and their extracted images. We should also be able to investigate the corresponding documents and their meta-data (keywords, authors, title, publisher date, page number, word distribution...).

3.3 Generation of topics and image clusters

Until now, the exploration system offers us only a rough overview of the document collection. Now it's time to take a closer look at its content. The software should have the possibility to determine the different topics of the documents. The goal is that we should be able to see how the topics differ from each other and which relevant documents fit more to certain topics than to others. Furthermore, the images of the documents should also play a major role in the investigation. For this purpose the software should create image clusters based on the similarity of all extracted images of the document collection. The whole process can be supported by defining an arbitrary number of filters to restrict the amount of used documents which should gives us the opportunity to find different topics or clusters depending on the meta-data of the documents.

3.3.1 Manual creation of topics

Since the automatic creation of topics might not always generate the expected results, the exploration system should also offer the possibility of manually creating a new topic. Since we are mainly dealing with text and image data, the user has two options: a text-based search and an image-based search. In the text-based version the user chooses the documents for the modeling process by applying a text query on all extracted documents. Then the system determines the significance of documents for the underlying query which gives the user the ability not only to see which documents are more significant given a certain text query but also to actually choose the documents the user wants to pass into the modeling process. On the other hand we have the image-based search which proposes the best fitting documents to the user based on uploaded query images or extracted images of specially selected documents.

3.3.2 Merging

Another optional feature is the merging of topics. We should be able to merge any number of topics into a single topic and combine all related data to not only get a possible better result but also to improve the evaluation process by bypassing limitations of one topic with the strengths of other topic or cluster data. The whole merge process should also be used for image clusters.

3.4 Relations between topics and image clusters

After we receive a detailed summary of the document collection based on generated topics and image clusters, the question arises whether we can find relationships between these two data accumulations which should also be provided by the exploration system. On top of that a weighting for each individual relation should be included, so that we can distinguish between bad and good connections. The goal is to find other topic related documents from the image cluster's relevant documents which are not included in the respectively appropriate topics.

Chapter 4

Implementation

4.1 Requirements

The key to a successful retrieval system is to choose the right descriptors that make the images as powerful and unique as possible. In other words we need a descriptor which combines colour and texture information to provide the best possible results. Therefore a primary goal of the project was to identify which descriptors are suitable for the different types of visualizations that may be implemented. Our research led us to FCTH (Fuzzy Color and Texture Histogram) and CEDD (Color and Edge Directivity Descriptor). Both are suitable for use in large image databases and are known as compact composite descriptors for content-based image retrieval. But in [PPK12], several experiments have been conducted and it has been observed that FCTH outperforms CEDD due to the incorporation of color and texture features in a single histogram which cannot precisely describe the image's complicated objects. Hence, CEDD cannot differentiate between different objects that have the same color and boundaries. That's the reason why the FCTH from the open source visual information retrieval library Lire is used [lir]. The last step was to find a proper application to extract the document's content (images, text and meta-data) of a document collection (PDF files). For this purpose we utilized the open source Java tool Apache PDFbox [pdf].

4.2 Reusability of extracted data

The extraction process of the document collection is a very computationally intensive procedure, hence requires a certain amount of time. The reason for this is not only the extraction of textual information but also images. It should be possible to store the extracted data on the hard drive so that it can be reloaded without much additional effort. For this purpose an import/export function is built into the exploration system where a separate folder will be created for each document in which a XML file serves as a storage medium for all kinds of textual data (text, meta-data, feature-vectors) whereas the document's images are stored separately in a sub folder. Each document is represented as a separate object within the program and contains all its own information. Since the whole information flow within the program is done through objects, the easiest way to export and import information is to save and load each document object as a serialised XML file. For this we use XStream [xst], a simple library to serialize objects to XML and back again.

4.3 Image Browser

The image browser is one of the user's first contact points to explore the extracted images of the documents. It displays all images that the system was able to extract. Furthermore, each image is assigned to a list of similar images. But how do we compare images? The similarity of two images will be determined by calculating the euclidean distance (also called L2 norm) of their 1-dimensional image feature vector histograms [FPR09]. The smaller the value, the more similar the images are. Assume we have two different images with the two feature vectors p and q, then the euclidean distance $d(p, q)$ between p and q will be calculated as followed:

$$d(p, q) = d(q, p) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

The respective result of the calculation is a list of consecutive images, displayed next to the start or query image, as shown in figure 4.1.

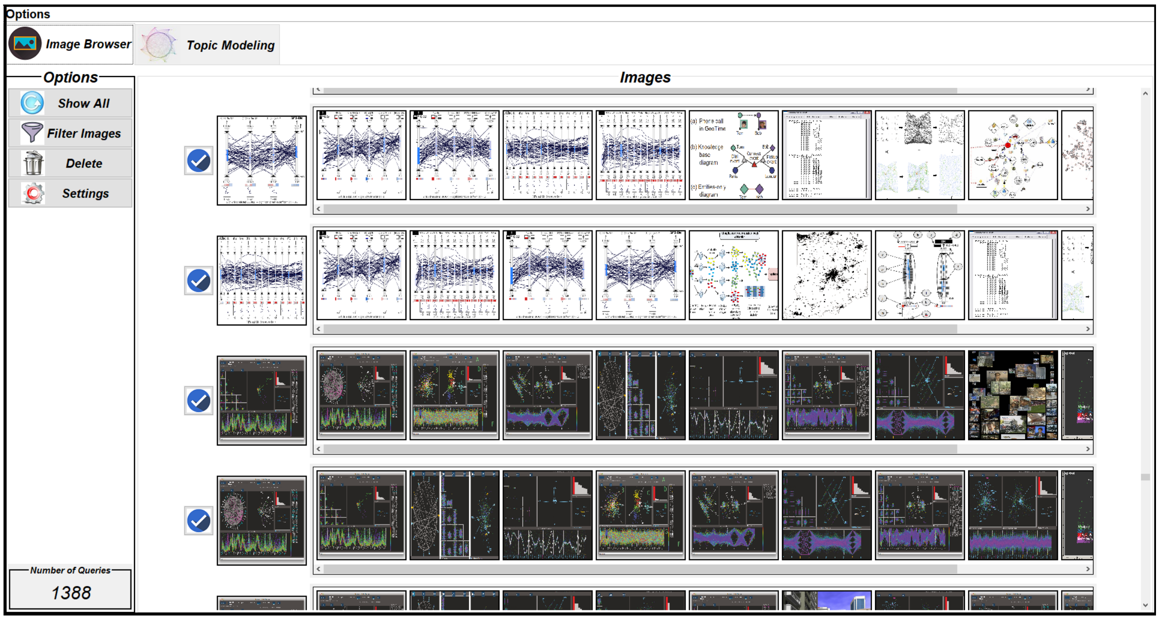


Figure 4.1: Extracted images and their most similar images

Furthermore, a special feature included in the image browser is the marking of images due to their corresponding meta-data (title, author, year and keywords). The marking itself is taken place by surrounding each image with a coloured border. Often it is useful to mark images after multiple criteria. As shown in figure 4.2, for this purpose the image border can be extended to actually allow us to mark images with up to six different colors. This results in a kind of color clustering which gives us a good opportunity to get a good first impression over the document collection. But the marking is not the only opportunity to investigate the underlying documents and their images. We can also add or remove filters to get even more separate image views.

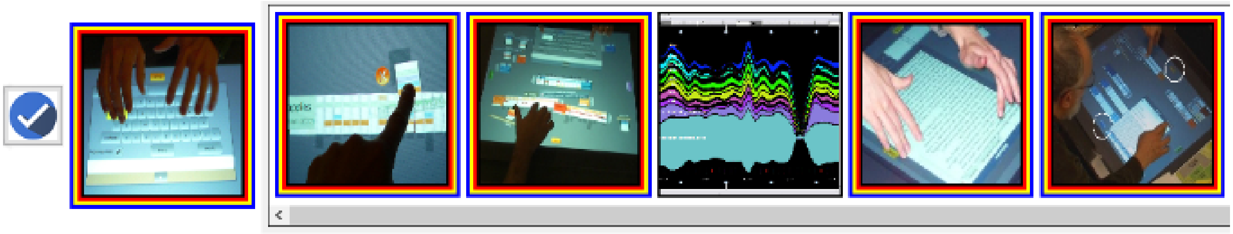


Figure 4.2: Overlapping markings in image browser

4.4 Document Browser

Beside the image browser, the document browser is the second direct interaction point between the user and the document collection. It visualizes all extracted documents as document cards, a very useful type of document visualization. First of all, the text and image information is processed separately. The text is used to generate a tag cloud to give the user a first impression of the textual content. The images are displayed in a grid. In addition, each document card is provided with its corresponding title, author, publisher date and number of pages. This data is used to identify each individual document. In order to give each document card an individual touch and to distinguish it from the others, it is colored by the dominant color of the images it contains. This is done by simply calculating the average RGB pixel values. In figure 4.3 an example of some document cards is shown.

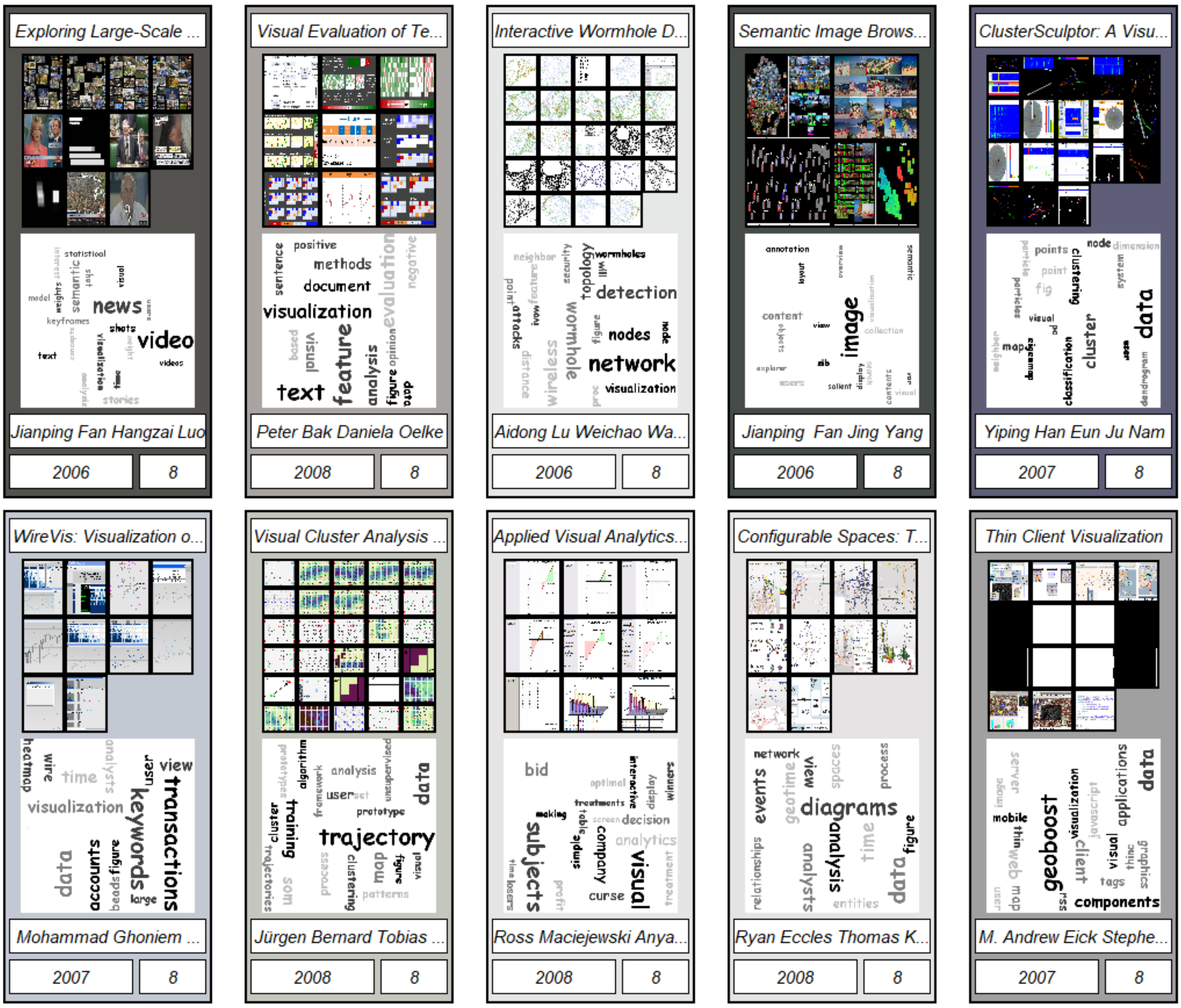


Figure 4.3: Example of the implemented Document Cards

The user is also provided with additional exploration tools: By right-clicking on a document card it can be opened by an external PDF viewer. It is also possible to filter the cards and get an extended view onto their contained images. For this purpose the software switches to the Image Browser to provide the possibility for further exploration.

4.5 Generation of topics and image clusters

In [XHLL16], the basic K-Means Algorithm is described as an unsupervised algorithm that is used to divide unlabeled data (data which is not defined in categories or groups) into K clusters based on some kind of similarity. At the beginning, each data object in the data set is defined as a single cluster where K data objects will be randomly selected as the initial clustering centers. After that, the distance of the remaining data objects to each of the K cluster centers will be calculated. Then, based on the nearest neighbour principle, each data object will be assigned to the nearest cluster (minimal distance to cluster center). At the end the new centroid of each cluster will be recalculated. The whole procedure will be repeated iteratively until the cluster partition is no longer changed or in other words no reassignment took place.

If we take a look onto that basic algorithm we recognise that we could actually use the same algorithm for our clustering purpose. The extracted images are defined as the above mentioned data objects and the similarity between different images, which is given by the euclidean distance, is used as a metric for the assignment to the nearest cluster. The only more complicate aspect is the recalculation of the new cluster centres. For this purpose we calculate the average distance of each image to every other image inside a cluster. The image with the smallest average distance will be set as the new centroid. From this we derived the following k-mean clustering algorithm:

1. Randomly select K images as the initial cluster centers

2. Calculate the euclidean distances from the remaining images to the initial cluster centers and categorize them according to the nearest one

3. Recalculate new cluster centroids by determining the cluster images with the smallest average euclidean distances to the current centres

4. Repeat step 2 and step 3 until convergence, e.g. no reassignment of images is taken place

As result we get the best K possible image clusters where each of the clusters will be displayed as a rectangle with a centralized image (representing the cluster centroid) surrounded by its top ten most similar images within the given cluster (as shown in figure 4.4).

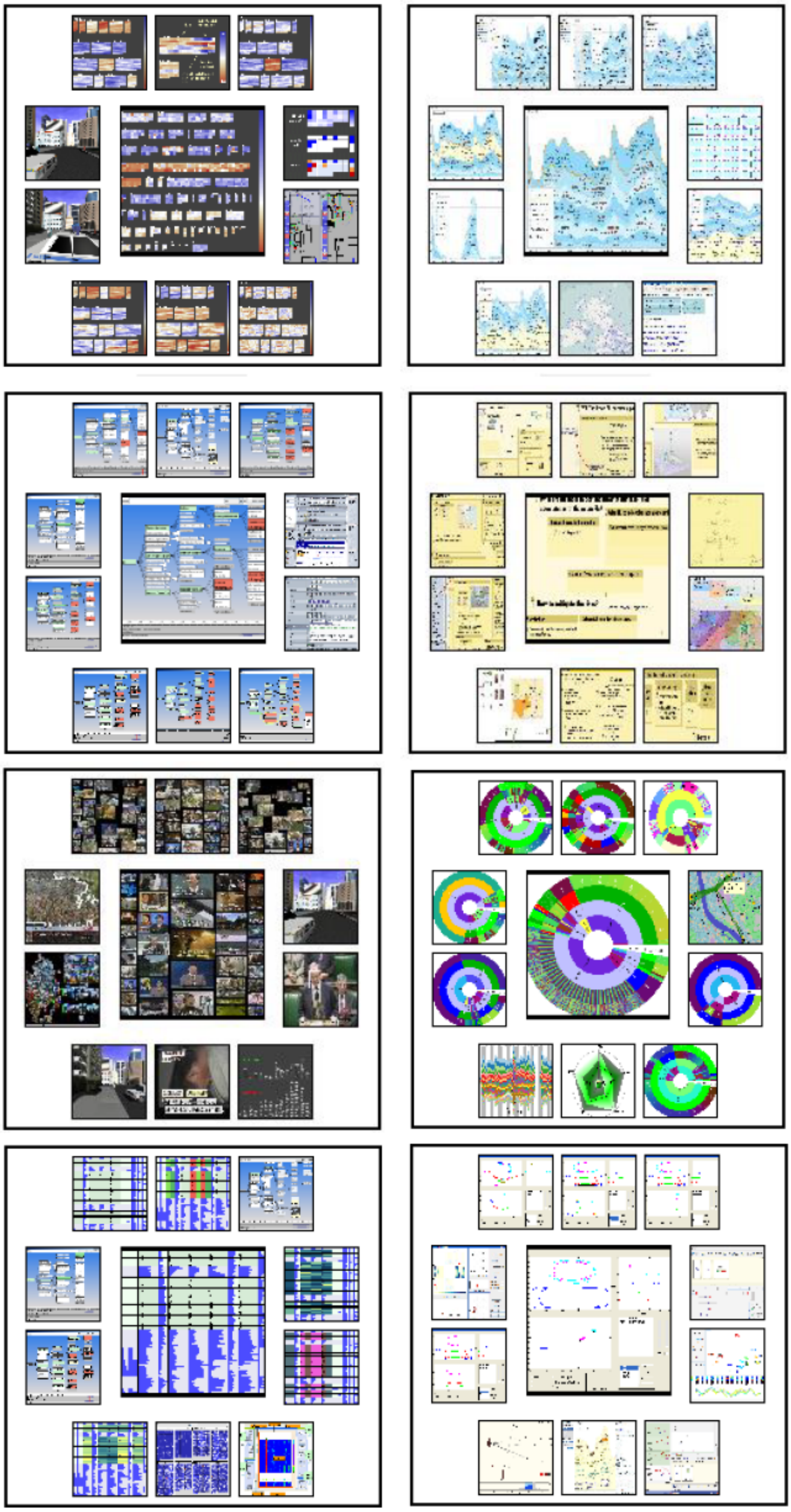
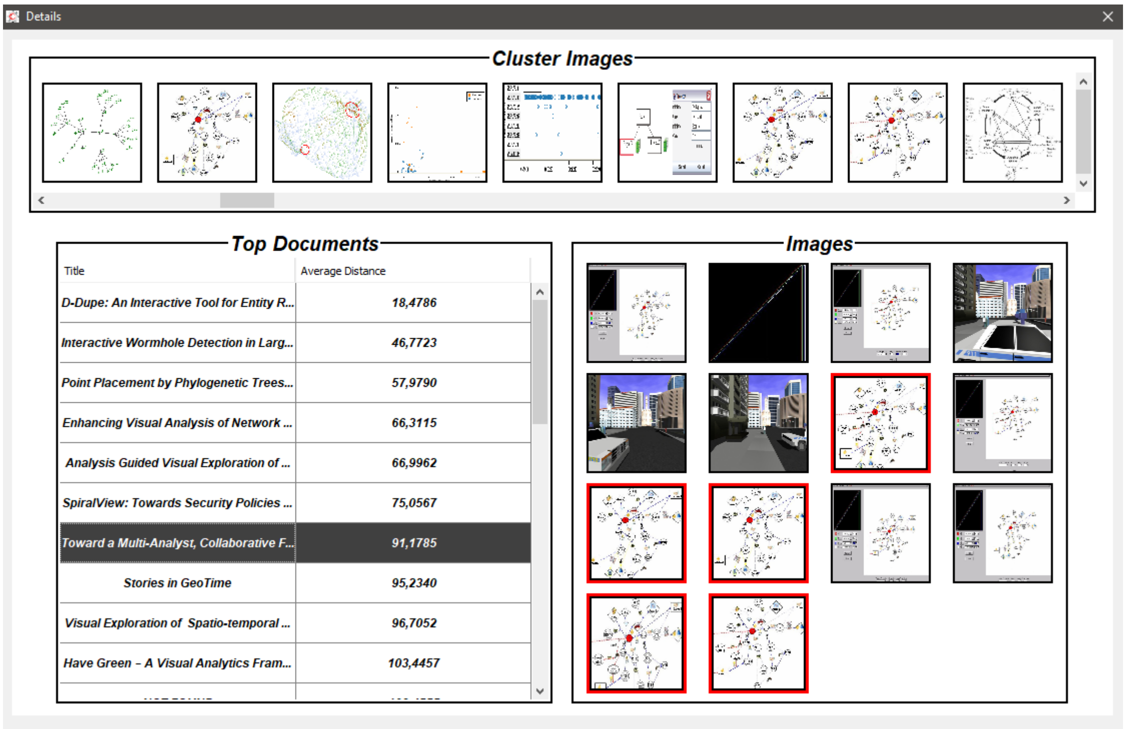


Figure 4.4: Examples of Image Clusters

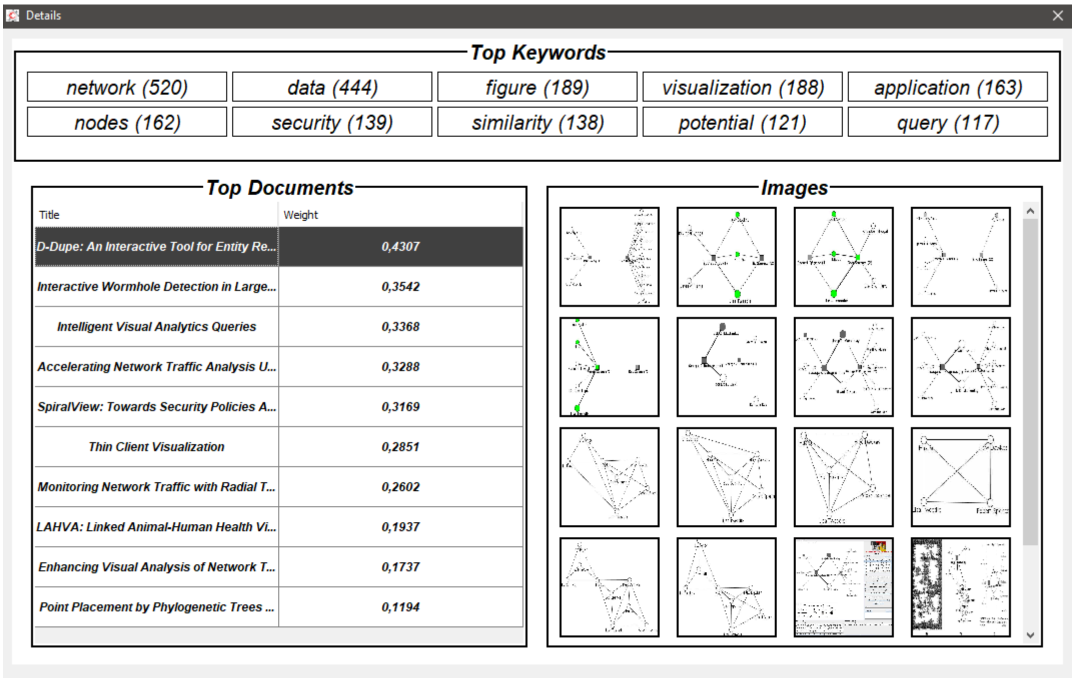
If we want to see more than just the most relevant images, we only need to click on the cluster. This will open a window where we can not only see all images inside that particular cluster, but also the most relevant documents that are representative for the cluster. The relevance is calculated by the average distance of all images inside a document to the centroid image of the cluster. By selecting one document, we get displayed all its contained images and those images that are actually contained in the cluster are highlighted in red (as shown in figure 4.5). Furthermore we are also able to actually open the paper with the underlying installed PDF viewer for further investigation.



Title	Average Distance
D-Dupe: An Interactive Tool for Entity R...	18,4786
Interactive Wormhole Detection in Larg...	46,7723
Point Placement by Phylogenetic Trees...	57,9790
Enhancing Visual Analysis of Network ...	66,3115
Analysis Guided Visual Exploration of ...	66,9962
SpiralView: Towards Security Policies ...	75,0567
Toward a Multi-Analyst, Collaborative F...	91,1785
Stories in GeoTime	95,2340
Visual Exploration of Spatio-temporal ...	96,7052
Have Green – A Visual Analytics Fram...	103,4457

Figure 4.5: Cluster details

Now after we found a way to successfully generate image clusters, the next step was to generate topics from the document's extracted texts. For this purpose we decided to use the Latent Dirichlet allocation (LDA) which is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words [BNJ03]. The java implementation of the LDA provided by Xianshun Chen [LDA] perfectly matches our conditions, hence we decided to use that package for our topic modelling. To have the software suggest the topics, the texts of the documents are passed to the external library as a list of strings, plus an additional constant that specifies the number of topics, which can be changed by the user at any time. But before we can actually start the modeling process, a further important step is needed: the preprocessing of the texts. Every text contains numbers or specific terms called stop words which usually refer to the most common words in a language. Since we are not interested in such terms, especially in topic modeling, every extracted text has to be preprocessed by filtering out those text segments. Another critical point is the stemming. Not every word is unique. Every word can be expressed in different forms (plural form, tenses, etc.). Our task now is grouping together the inflected forms of a word into a single term so that it can be analysed and identified by its dictionary form. This process is called stemming or lemmatization. For this purpose we use the external library Porter-Stemmer [PST] which supports that kind of term normalisation by using the Porter Stemming Algorithm [SJW97]. After prepocessing our texts, we pass them to our topic generation library which provides us a list of topics. Each topic is represented by its most relevant words and the corresponding most relevant documents ranked by their affiliation weight. To visualize the topic details, we implemented a visualization option for inspecting a topic by clicking on it. As shown in figure 4.6, the software opens a window where all the mentioned details will be displayed, including the possibility to investigate the most relevant documents themselves. By selecting one of the shown documents, the same features as in the cluster investigation can be performed.


Details

Top Keywords

network (520)	data (444)	figure (189)	visualization (188)	application (163)
nodes (162)	security (139)	similarity (138)	potential (121)	query (117)

Top Documents

Title	Weight
D-Dupe: An Interactive Tool for Entity Re...	0,4307
Interactive Wormhole Detection in Large...	0,3542
Intelligent Visual Analytics Queries	0,3368
Accelerating Network Traffic Analysis U...	0,3288
SpiralView: Towards Security Policies A...	0,3169
Thin Client Visualization	0,2851
Monitoring Network Traffic with Radial T...	0,2602
LAHVA: Linked Animal-Human Health Vi...	0,1937
Enhancing Visual Analysis of Network T...	0,1737
Point Placement by Phylogenetic Trees ...	0,1194

Images

Figure 4.6: Topic details

4.5.1 Manual creation of topics

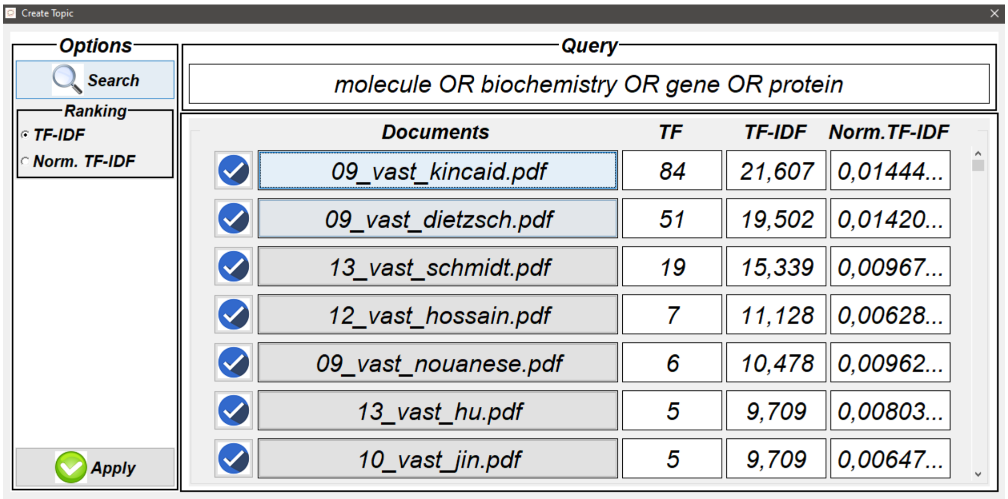
To manually create topics, we only have to use the already mentioned java library of the Latent Dirichlet Allocation. The only question we have to ask ourselves is: What documents do we need? Obviously we cannot use documents arbitrarily for the generation, because it should be a useful topic after all. For this reason we provide two different functions for the specified selection of documents. The text-based and image-based search. The text-based search uses a query to determine the most relevant documents. The query itself can also contain several different terms to make the search as precise as possible. But how do we calculate the relevance of documents based on the query? TF-IDF is one of the most important measures when we are talking about automated text analysis, document search and information retrieval. It is a statistical measure that can provide the relevance of documents inside a collection. In general it consist of two metrics: the Term Frequency (TF, occurrences of a word inside a document) and the Inverse Document Frequency (IDF, how common is a word with regard to an entire set of documents). Assume in the document collection D, the overall number of documents is N and the number of documents in which a term t appears is n_t, the IDF is calculated as follows:

$$idf(t, D) = \log\left(\frac{N}{1 + n_t}\right)$$

By multiplying the two metrics we get the TF-IDF score (d represents a single document).

$$tf\text{-}idf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

Now we face the problem that this approach doesn't include the fact that certain documents are larger than others, therefore contain much more words and have a higher probability to actually contain a search term several times. To address this problem we include the document size by offering a normalization method where we divide the TF-IDF score by the overall number of words inside a document (words that occur several times count as one). This whole calculation procedure will be done for every document which gives us a list of papers ranked by their relevance based on the given text query (example is shown in figure 4.7).



Ranking
● TF-IDF
○ Norm. TF-IDF

Query
molecule OR biochemistry OR gene OR protein

	Documents	TF	TF-IDF	Norm.TF-IDF
✓	09_vast_kincaid.pdf	84	21,607	0,01444...
✓	09_vast_dietzsch.pdf	51	19,502	0,01420...
✓	13_vast_schmidt.pdf	19	15,339	0,00967...
✓	12_vast_hossain.pdf	7	11,128	0,00628...
✓	09_vast_nouanese.pdf	6	10,478	0,00962...
✓	13_vast_hu.pdf	5	9,709	0,00803...
✓	10_vast_jin.pdf	5	9,709	0,00647...

Figure 4.7: Text query with the corresponding document results, ranked by TF-IDF

Now we still have the possibility to inspect the displayed documents by simply clicking onto a document. A window will be opened where further document specific information will be displayed as well as a link to open the respective document with an installed document viewer. Then we can choose the documents which should be used to generate a topic. If we are satisfied with the choice, a simple click on the apply button is enough to complete the process and the topic is displayed on the topic section.

Alternatively, as described above we can also generate a topic using the image based search. For this purpose we can either load predefined images from a folder or extract them from previously collected documents. Those images are then displayed by the exploration system (shown in figure 4.8) and we have the possibility to remove unsuitable images to support the image based search.

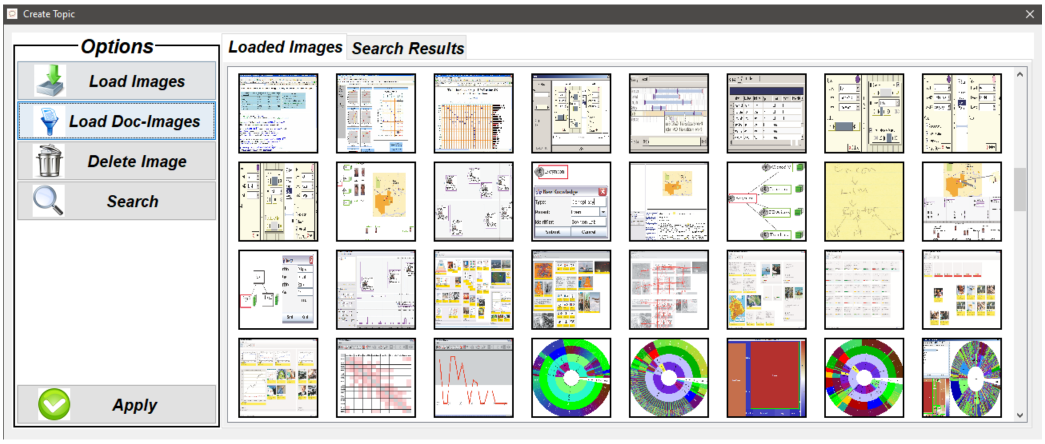
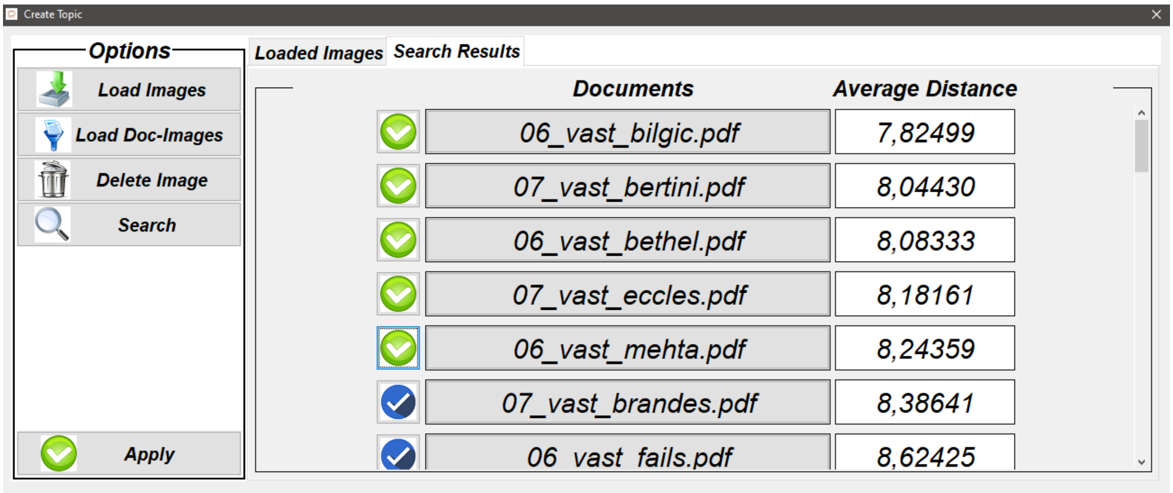


Figure 4.8: Displayed query images

The ranking itself is done by calculating the average euclidean distance between the displayed query images and the documents images from the collection. As a result we get a list of most relevant documents which again can be further investigated by either inspect their additional details or open them with a external PDF viewer. Finally, as shown in figure 4.9, we can again select the documents we want to use for generating the topic.



	Documents	Average Distance
✓	06_vast_bilgic.pdf	7,82499
✓	07_vast_bertini.pdf	8,04430
✓	06_vast_bethel.pdf	8,08333
✓	07_vast_eccles.pdf	8,18161
✓	06_vast_mehta.pdf	8,24359
✓	07_vast_brandes.pdf	8,38641
✓	06_vast_fails.pdf	8,62425

Figure 4.9: Result of an image based query, ranked by the average euclidean distance

4.5.2 Merging

In our exploration system there are two data sets which can be merged: topic models and image clusters. Let us start with the relatively simple ones, the image clusters. Each cluster consists of a number of images. In order to merge two or more clusters, all of their images must first be used to calculate the centroid. This is done by calculating the average euclidean distance of each image to all other images in the cluster. Once this part is done, the last step is to determine the most relevant documents by the number of images present in the cluster and their distance to the centroid. After that the merge process is successfully completed. Merging topic models is a little bit more complicated since there is more data that has to be merged. As we already know, a topic contains a set of its most relevant words and the corresponding weights (the same goes for its most relevant documents). Therefore we must create an intersection of word lists of all involved topics. In the case of dealing with equal words, the corresponding weights will be added and in the end each weight will be divided by the total number of topics that are chosen to be merged. The same procedure will be done for the document set. The result is a new topic with its two new accumulated data sets and their new corresponding weights.

4.6 Relations between topics and image clusters

To be able to calculate relations between topics and image cluster we first need to know how do we define our relations. In our case, relations are found through the common relevant documents. In other words we create an intersection for both, the cluster's and the topic's relevant document sets and if the result size is greater zero we found a connection. This approach will be done for every possible constellation between topics and clusters. Additionally a weighting will be assigned to each relation to distinguish between good and less good relations. The calculation is done by using the jaccard similarity coefficient which measures the similarity between finite sample sets. It is defined as the size of the intersection divided by the size of the union of the sample sets [NSNW13]. So let A, B be two document sets then we calculate the jaccard similarity coefficient J as displayed below.

$$J(A,B) = \frac{(|A \cap B|)}{(|A \cup B|)} = \frac{(|A \cap B|)}{(|A| + |B| - |A \cap B|)}$$

The next step is to extend the jaccard similarity coefficient dividing the results by the maximal calculated coefficient. The result is a similarity distribution which in our case involves three different subdivisions:

Relations	Extended Jaccard Coefficient	Color
Best	[0.75 - 1.00]	■
Average	[0.50 - 0.75[	▓
Bad	[0.25 - 0.50[	░

The whole procedure provides us a possibility to not only graphically distinct between good and less good relations (as shown in figure 4.10) but also to only display relations that we actually want.

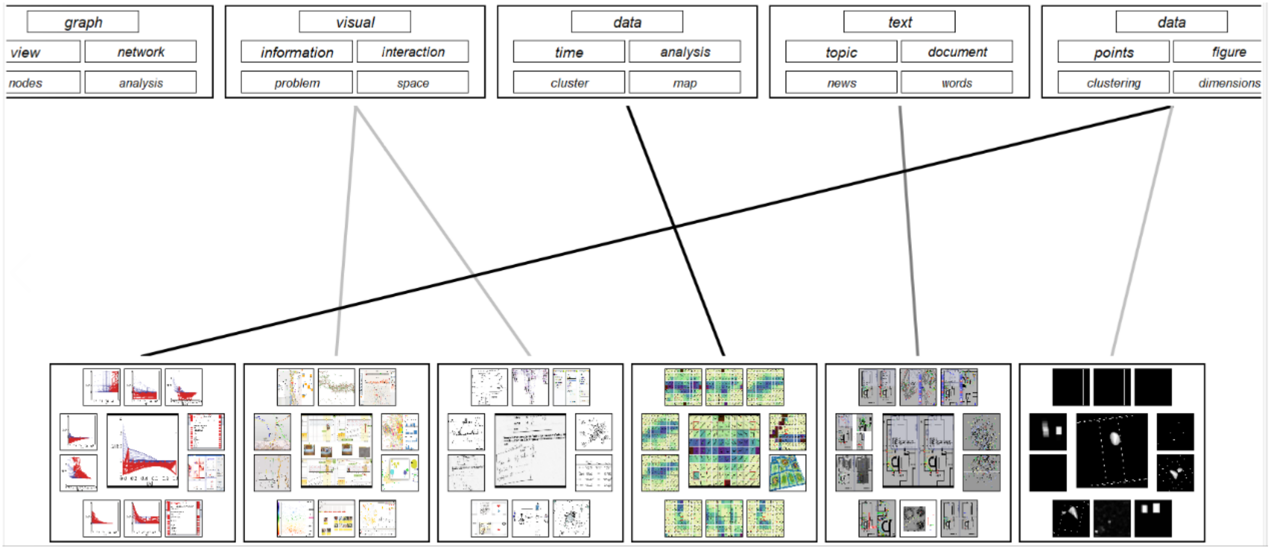
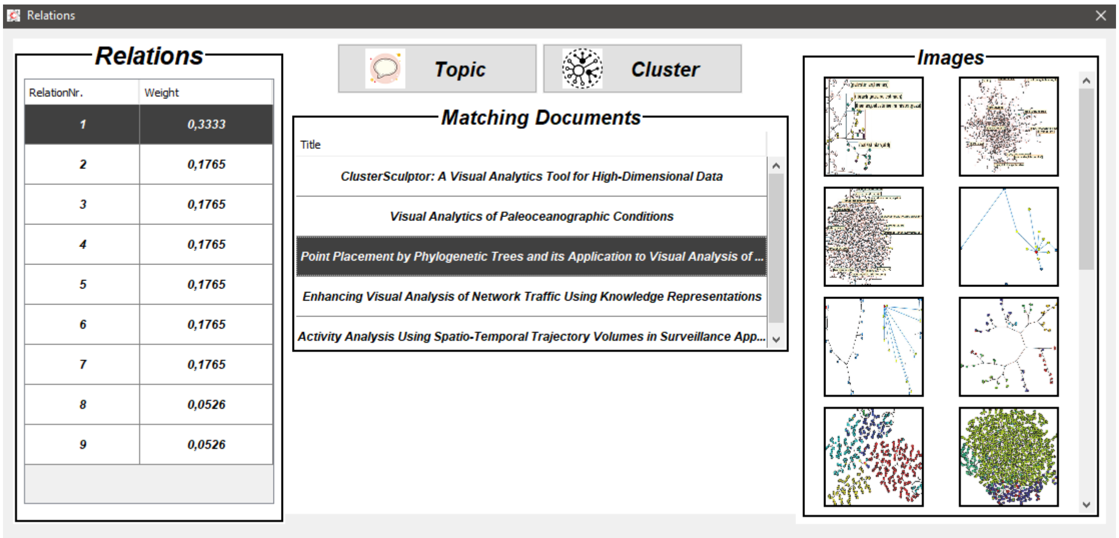


Figure 4.10: Display all relations between topics and clusters

In addition to creating the relations we are also able to investigate them. For this purpose the exploration system offers a relation investigation option where we only have to select a topic or cluster and, as shown in figure 4.11, the system displays all the involved relations ranked by their weight (jaccard similarity coefficient). Furthermore we can actually go through every relation and see the involved topic and image cluster as well as all matching documents (inclusive their contained images) which again can be opened by a external PDF viewer.



Relations

RelationNr.	Weight
1	0,3333
2	0,1765
3	0,1765
4	0,1765
5	0,1765
6	0,1765
7	0,1765
8	0,0526
9	0,0526

Matching Documents

Title
ClusterSculptor: A Visual Analytics Tool for High-Dimensional Data
Visual Analytics of Paleoceanographic Conditions
Point Placement by Phylogenetic Trees and its Application to Visual Analysis of ...
Enhancing Visual Analysis of Network Traffic Using Knowledge Representations
Activity Analysis Using Spatio-Temporal Trajectory Volumes in Surveillance App...

Figure 4.11: A detailed view of all relations from a topic/cluster

4.7 Extended Overview

The exploration tool also offers a possibility to give a more accurate overview over the topic and image cluster network. For this purpose the software opens a new window in which the described network will be displayed from a whole new perspective. The size of the window adapts dynamically to the used screen size and the image clusters are displayed as image grids that show all contained images, as shown in figure 4.12.

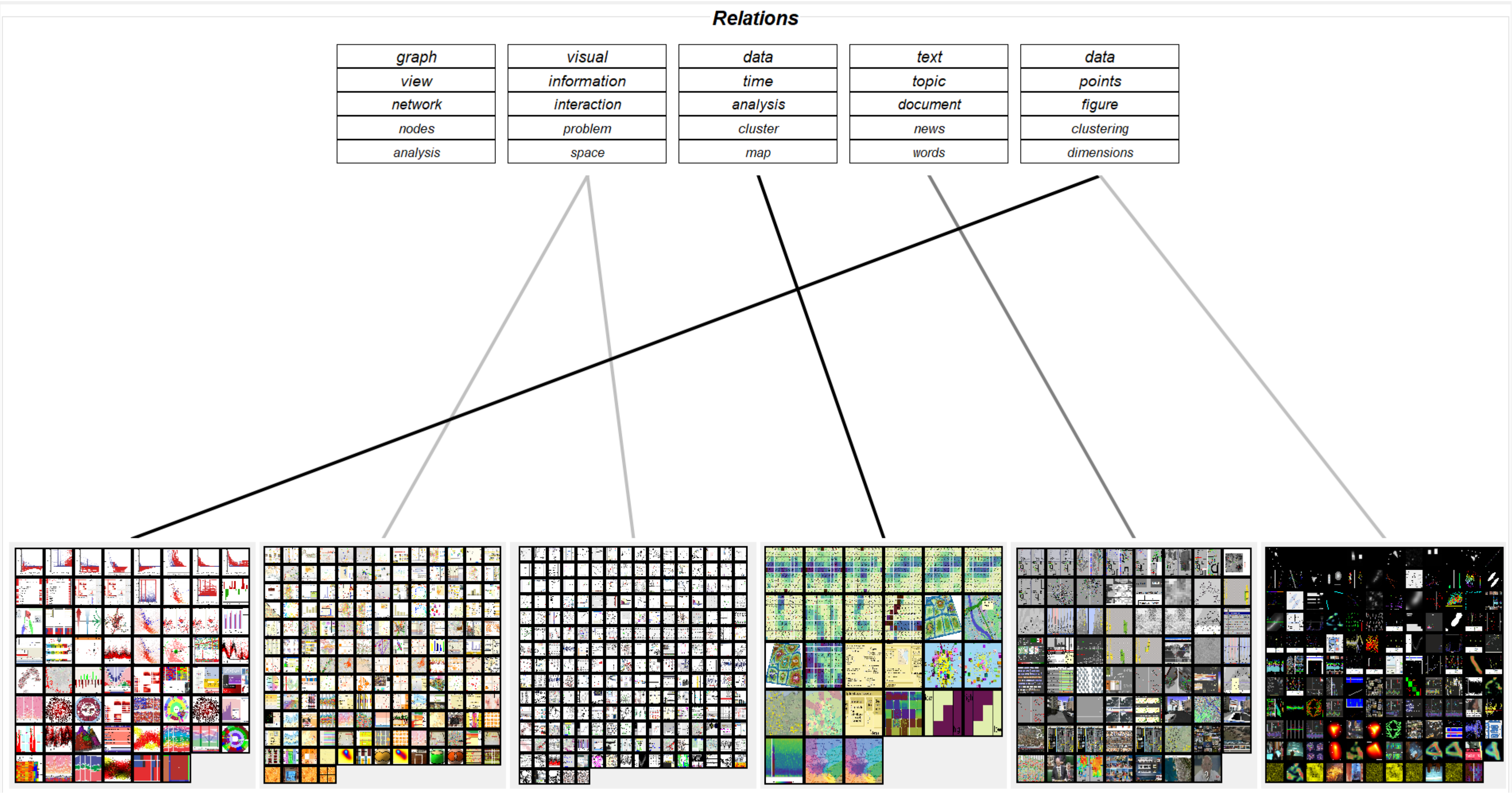


Figure 4.12: Extended overview over a topic/cluster network

Chapter 5

Evaluation

5.1 Usecase: Automatic creation of Topics and Clusters

5.1.1 Description

The software project provides a feature where the user can create a self defined number of topics and clusters which can be freely investigated. The whole process can be supported by defining an arbitrary number of filters to restrict the amount of used documents which gives the user the opportunity to find different topics or clusters depending on the meta-data of the documents.

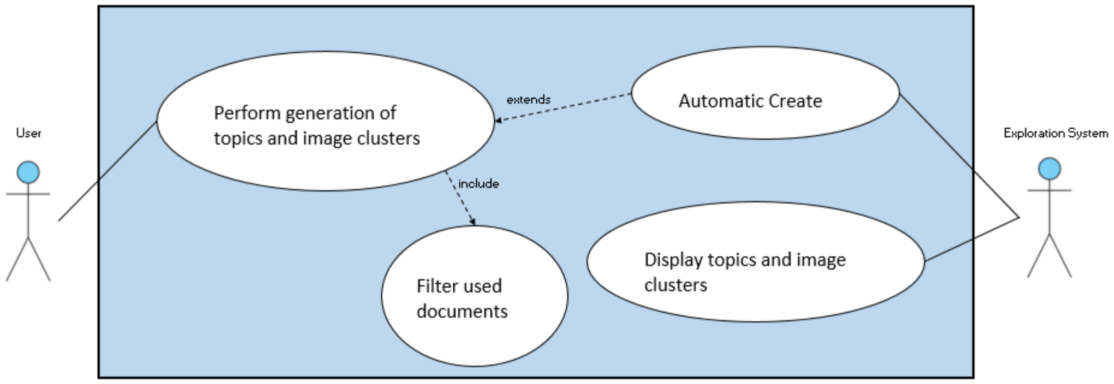

User
Perform generation of topics and image clusters
extends
Automatic Create
include
Filter used documents
Display topics and image clusters
Exploration System

Figure 5.1: Use Case Diagram: Automatic creation of topics and image clusters

5.1.2 Main Scenario: Create Topics and Clusters

1. Extract or import data from the document collection
2. User opens topic model tab
3. User sets number of topics/clusters in settings
4. User presses “Generate”-Button
5. Systems displays window with optional filter options for using documents in the modelling process
6. User states some filters (if needed) and presses the “Start”-Button
7. System generates and displays topics (boxes with the five most important keywords inside the topic) and image clusters (boxes with a center image surrounded by its most similar images)

8. User right-clicks on topic/cluster and clicks on “Show Details” in pop-up menu

9. Depending whether it’s a topic or a cluster, the system shows the following details:

   (a) Topic

      i. Top keywords and their corresponding weight
      ii. Top documents and their corresponding weight
      iii. Images of a selected top document

   (b) Cluster

      i. All images in the cluster
      ii. Top documents and their corresponding weight
      iii. Images of a selected top document

10. User selects a document of his choice

11. The system displays all images from that document (in case of a cluster, the images which are also contained in the cluster will be marked)

5.1.3 Alternative Scenario 1: Topic/Cluster Modelling with predefined Filters from Image Browser

1. Extract or import Data from a document collection

2. User defines filter in image browser until he is satisfied with the displayed query images

3. User opens topic model tab

4. User sets number of topics/clusters in settings

5. User presses “Generate”-Button

6. Systems displays window with optional filter options for using documents in the modelling process

7. User presses “Copy Filters from Image Browser”

8. System copies used filters from image browser to the topic modelling process

9. User presses “Start”-Button

10. System generates and displays topics/clusters

5.1.4 Alternative Scenario 2: Topic/Cluster Modelling with containing documents of a certain Topic or Cluster

1. Extract or import Data from a document collection
2. User opens Topic Model Tab
3. User sets number of Topics/Clusters in Settings
4. User presses “Generate” Button
5. User states some filters (if needed) and presses “Start”
6. System generates and displays topics and image clusters
7. User right-clicks on a topic/cluster and clicks on “Modelling with containing documents”
8. System creates and displays topics/clusters, but only out of the top documents inside of the specific topic/cluster

5.1.5 Finding

This finding deals with the question of finding any kind of meaningful image clusters that exist in certain document collections. With image clustering the user is not only able to group similar images together but also to identify used applications during the scientific work of the concerning papers. One of the best examples is the obtained image cluster shown in 5.2.

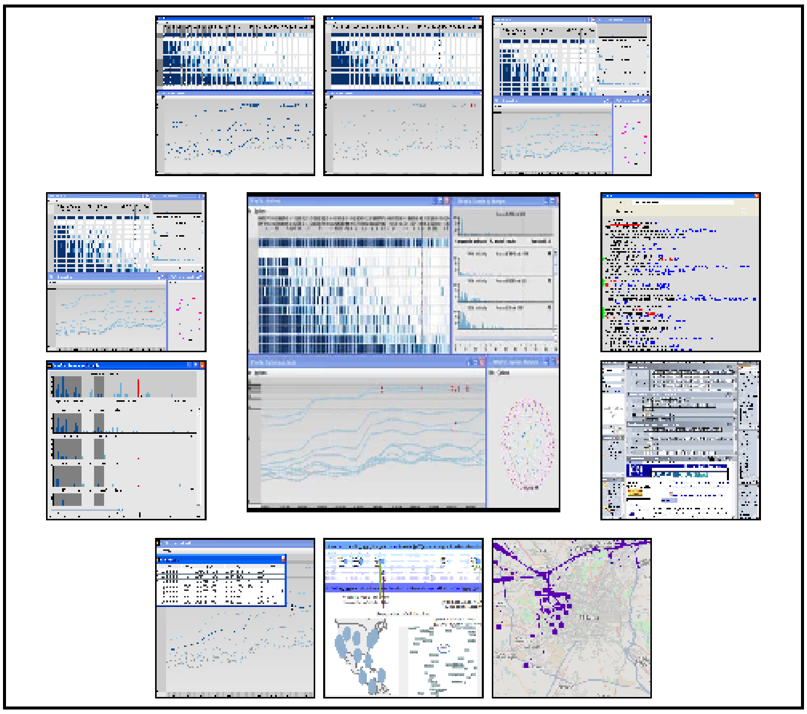
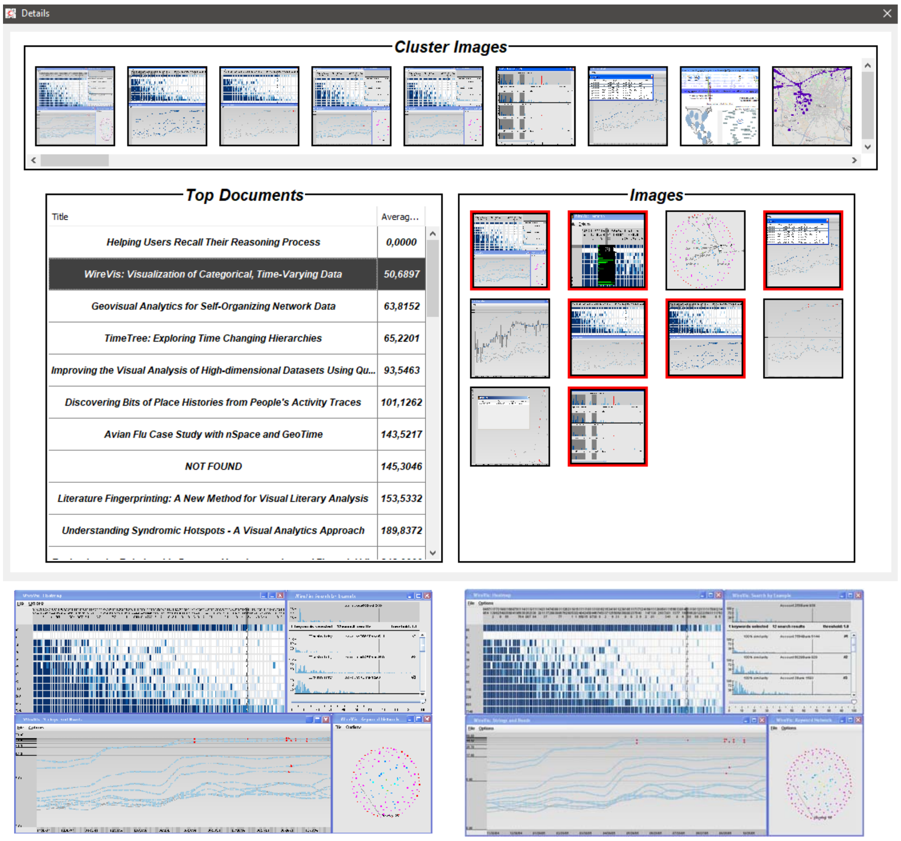


Figure 5.2: Cluster with images showing the WireVis application

We can see a couple of images showing the usage of WireVis, a multiview approach that assists analysts in exploring large numbers of categorical, timevarying data containing wire transactions [CGK*07]. In further investigation of the mentioned cluster we also can see that not only one but three papers were found containing images of WireVis: In [CGK*07] they present WireVis as a whole, describing the method as highly interactive and as a combination of a keyword network view, a heatmap and a search-by-example tool which gives the user a global overview of the data by aggregating and organize groups of transactions for better investigation and analysis of individual records while in [JWL*08] and [LSD*10] WireVis was used in order to understand the user's reasoning process through his interactions during a visual analysis.



Title	Averag...
Helping Users Recall Their Reasoning Process	0,0000
WireVis: Visualization of Categorical, Time-Varying Data	50,6897
Geovisual Analytics for Self-Organizing Network Data	63,8152
TimeTree: Exploring Time Changing Hierarchies	65,2201
Improving the Visual Analysis of High-dimensional Datasets Using Qu...	93,5463
Discovering Bits of Place Histories from People's Activity Traces	101,1262
Avian Flu Case Study with nSpace and GeoTime	143,5217
NOT FOUND	145,3046
Literature Fingerprinting: A New Method for Visual Literary Analysis	153,5332
Understanding Syndromic Hotspots - A Visual Analytics Approach	189,8372

Figure 5.3: The big picture shows all WireVis images of the main paper [CGK*07] and the two small pictures, which on closer look are the same, were found in the two other papers ([JWL*08], [LSD*10])

5.2 Usecase: Creation of user specified topics

5.2.1 Description

The exploration system offers the possibility of manually creating a new topic by choosing the required documents which will be included in the generation of the topic. Hereby the software comes up with two options: a text-based search and an image-based search. In the text-based version the user chooses the documents for the modeling process by applying a text query to all documents. On the other hand we have the image-based search which proposes the best fitting documents to the user based on uploaded query images or extracted images of specially selected documents.

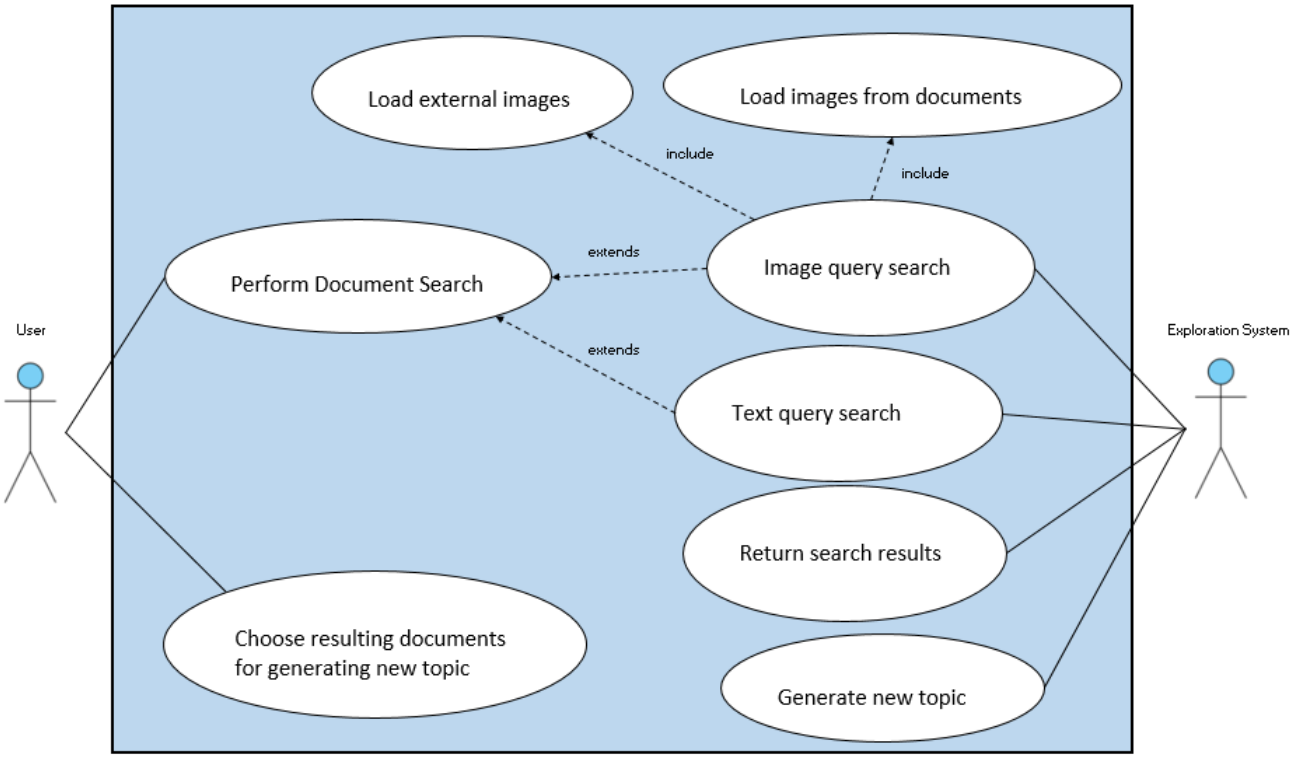


Figure 5.4: Use Case Diagram: Manually creating topics with text/image query search

5.2.2 Main Scenario: Create topic with text query

1. User opens Topic Model Tab with already calculated Topics/Clusters
2. User presses “Create Topic”- Button
3. System opens “Create-Topic”-Window where the user can choose on which criteria a new Topic will be generated (text-based or image-based query)
4. User chooses the text-based creation of a topic and the system opens the according window
5. User inserts a text query (can combine several queries with the logical operator OR)
6. User starts search
7. System displays documents ranked by their relevance (ranked by either TF-IDF or normalized TF-IDF)

8. User can click on one of the documents to show some additional information

9. System displays details of the document including a link to open it

10. User chooses a number of documents and press “OK”

11. System generates one Topic out of those documents and appends it to the list of existing Topics

5.2.3 Alternative Scenario: Create Topic with Image Query

1. User opens Topic Model Tab with already calculated Topics/Clusters

2. User presses “Create Topic”- Button

3. System opens “Create-Topic”-Window where the user can choose on which criteria a new Topic will be generated (text-based or image-based query)

4. User chooses the image-based creation of a topic and the system opens the according window

5. User can choose one of the two possibilities:

   (a) Load images from a folder

   (b) Load documents from a folder and the exploration software extracts the contained images

6. System displays images

7. User can remove certain images from the query if he wants to

8. User agrees with the image query and starts the search

9. System displays documents ranked by their relevance

10. User can click on one of the documents to show some additional information

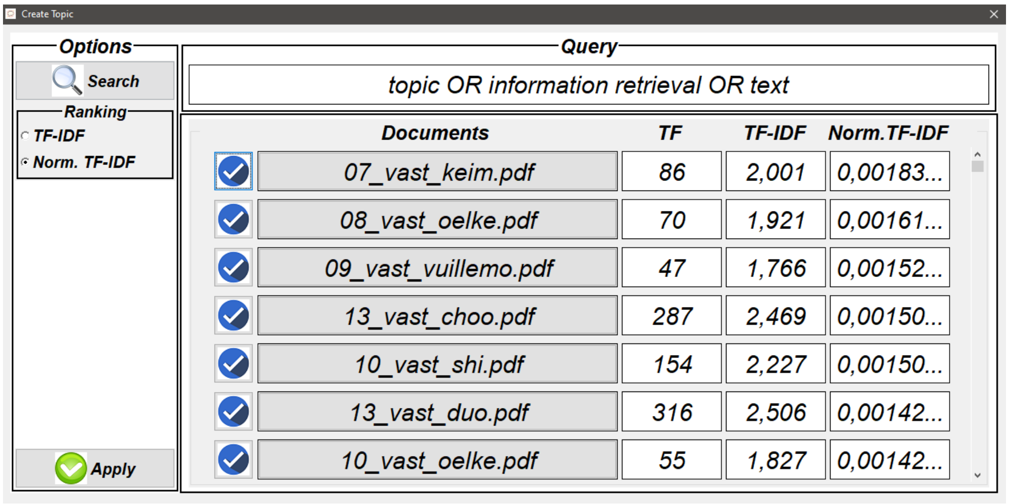
11. System displays details of the document including a link to open it

12. User chooses a number of documents and press “OK”

13. System generates one Topic out of those documents and appends it to the list of existing Topics

5.2.4 Finding

As we are dealing with two different topic generating methods we also want to know which of them is the best one. The first topic where we wanted to test this comparison was a subarea of this master thesis itself: text based information retrieval. To manually generate such a topic we created it using a text query, as shown in 5.5.


Create Topic

Options

Search

Ranking

○ TF-IDF

● Norm. TF-IDF

Apply

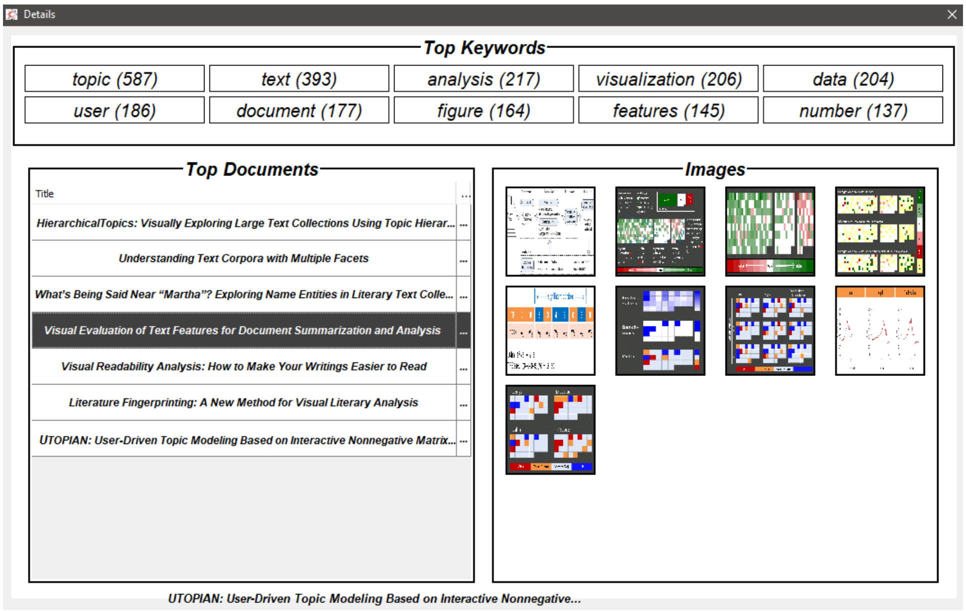
Query

topic OR information retrieval OR text

	Documents	TF	TF-IDF	Norm.TF-IDF
☑	07_vast_keim.pdf	86	2,001	0,00183...
☑	08_vast_oelke.pdf	70	1,921	0,00161...
☑	09_vast_vuillemo.pdf	47	1,766	0,00152...
☑	13_vast_choo.pdf	287	2,469	0,00150...
☑	10_vast_shi.pdf	154	2,227	0,00150...
☑	13_vast_duo.pdf	316	2,506	0,00142...
☑	10_vast_oelke.pdf	55	1,827	0,00142...

Figure 5.5: Most relevant documents based on the given text query

Through this approach the software suggested us the most relevant documents concerning the given query. The result was the expected well ordered topic which exactly deals with different text analysis content, for example: text understanding methods, readability analysis, literature fingerprinting, hierarchical topics and an application called UTOPIAN (User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization), as shown in 5.6. Now it was time to actually do the same but with image-based search. We searched for some content related documents in the EEE Xplore digital library and passed it to the exploration system. I transferred the documents to the software which then extracted the contained images. But the result was sobering. By comparing all the images in the respective documents, we recognized a big discrepancy between relatively good and not well matched images, in other words, we got documents which didn't fit the actual topic and therefore we couldn't hardly create topics based on image search alone. Therefore we can assume that if images differ too much in their actual background color (e.g. white and black background), it is very hard to find good results using the current image descriptor.


Details

Top Keywords

topic (587)	text (393)	analysis (217)	visualization (206)	data (204)
user (186)	document (177)	figure (164)	features (145)	number (137)

Top Documents

Title

HierarchicalTopics: Visually Exploring Large Text Collections Using Topic Hierar...

Understanding Text Corpora with Multiple Facets

What's Being Said Near "Martha"? Exploring Name Entities in Literary Text Colle...

Visual Evaluation of Text Features for Document Summarization and Analysis

Visual Readability Analysis: How to Make Your Writings Easier to Read

Literature Fingerprinting: A New Method for Visual Literary Analysis

UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix...

Images

UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative...

Figure 5.6: Manual created topic based on a text query

On the other hand if the query images don't have big differences if we look on the overall background in general, then we actually get pretty descent results. An example of such a search query is shown in figure 5.7. In this case we used six network images with a neutral white background color as query images.

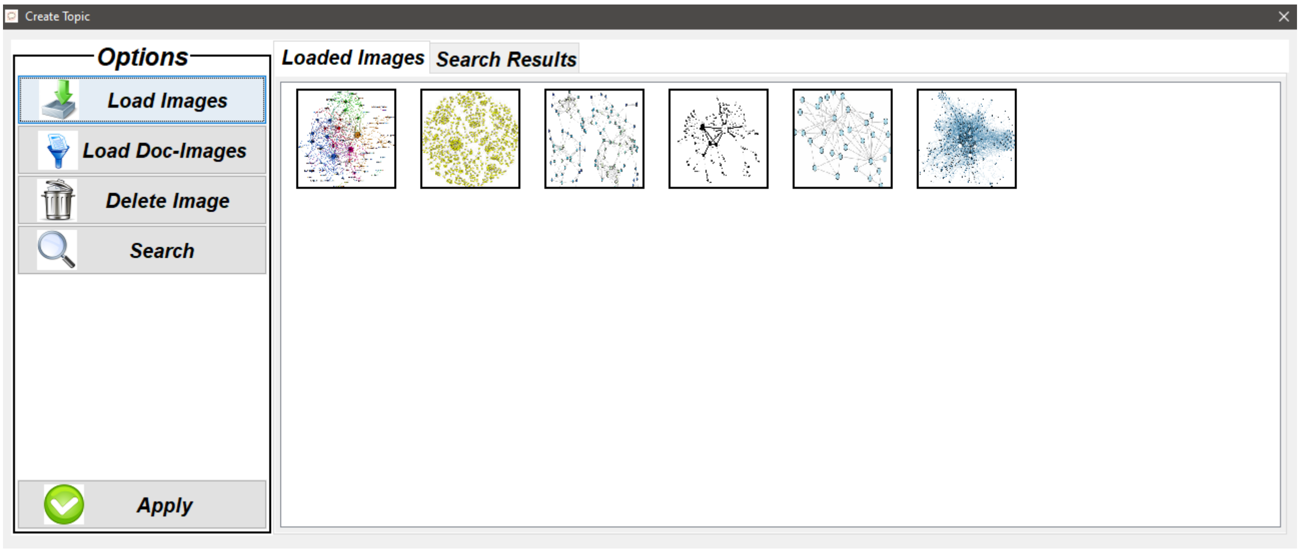
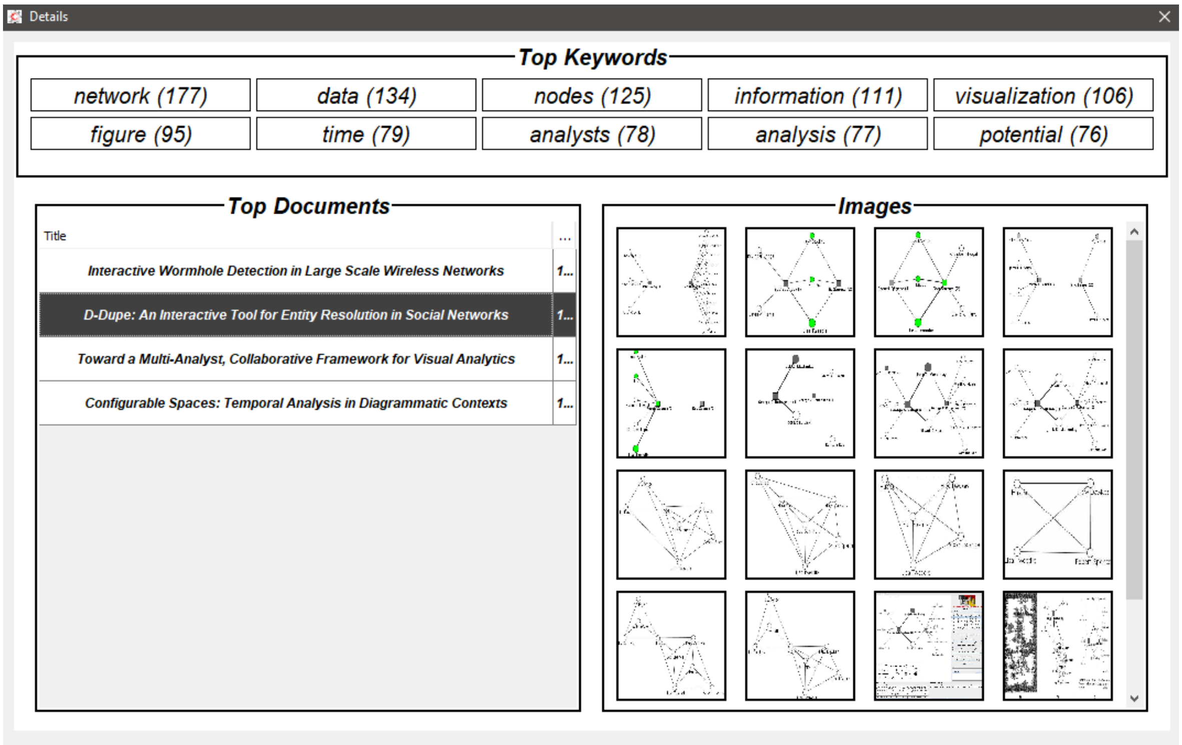


Figure 5.7: Query images of a network

Compared to other queries, as we can see in figure 5.8, we got a very good result by obtaining a topic that actually deals with networks.



network (177)	data (134)	nodes (125)	information (111)	visualization (106)
figure (95)	time (79)	analysts (78)	analysis (77)	potential (76)

Title	...
Interactive Wormhole Detection in Large Scale Wireless Networks	1...
D-Dupe: An Interactive Tool for Entity Resolution in Social Networks	1...
Toward a Multi-Analyst, Collaborative Framework for Visual Analytics	1...
Configurable Spaces: Temporal Analysis in Diagrammatic Contexts	1...

Figure 5.8: Manual created topic based on the network images

5.3 Usecase: Merging Topics and Clusters

5.3.1 Description

After an automatic generation of topics and clusters, the user likely can't find proper relations among them. Hence merging an arbitrary number of topics/clusters should not only give the user the opportunity to narrow down the possibilities of relations and therefore a potential boost of better results but also to improve an evaluation by bypassing limitations of one topic/cluster with the strengths of other topic/cluster data.

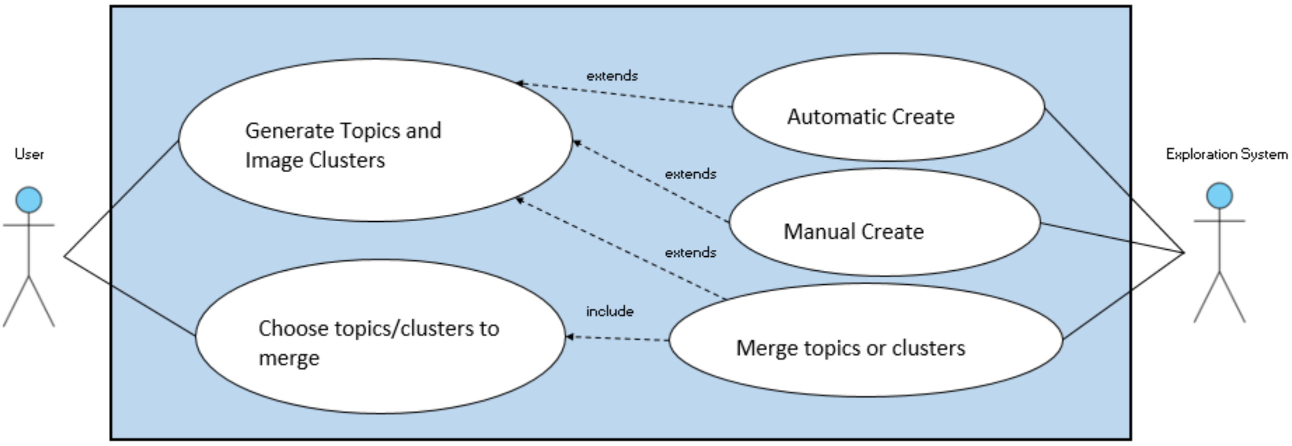


Figure 5.9: Use Case Diagram: Merging topics or clusters

5.3.2 Main Scenario

1. Extract or Import Data from a document collection

2. User opens Topic Model Tab

3. User sets number of Topics/Clusters in Settings

4. User presses "Generate" Button

5. Systems displays window with optional Filter Options (for used documents in the modelling process)

6. User states some filters (if needed) and presses "Start"

7. System generates and displays topics (boxes with the 5 most important keywords inside the topic) and image clusters (boxes with a center image surrounded by its most similar images)

8. User is not satisfied with the found clusters/topics and investigates them to retrieve information of his/her interest and selects a couple of them. Then the user presses the "Merge"-Button

9. The System merges the Topics/Clusters to one Topic/Cluster and appends it at the end of the existing/remaining Topics/Clusters

5.3.3 Finding

If we deal with topics and image clusters, it is not often guaranteed that we also find good relations among them. Therefore the exploration system offers a merging function to bring one or more topics/clusters together and combine their contained information. The result is not only a potentially better topic but the user is also capable of finding new topic related documents which wasn't necessarily possible in the previous not merged state of the topics. An example is shown in 5.10, where we can see two topics where no good relations were found.

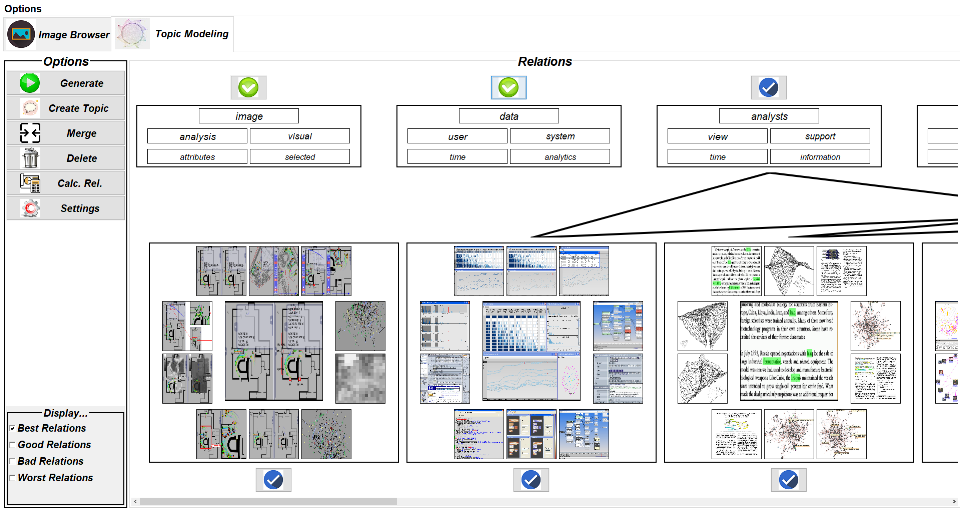


Figure 5.10: Topics with no good relations

As a consequence we decided to merge them together. After the merge process, the new topic was displayed at the end of the topic list and the relations were calculated again. As we can see in figure 5.11, this time two good relations were found. Now the first step was to investigate the new topic to actually identify which kind of topic we were dealing with (this process is shown in figure 5.12). The five most relevant keywords were: data, image, analysis, user and system. We also checked some of the most relevant documents of that topic to get a brief overview about the textual content.

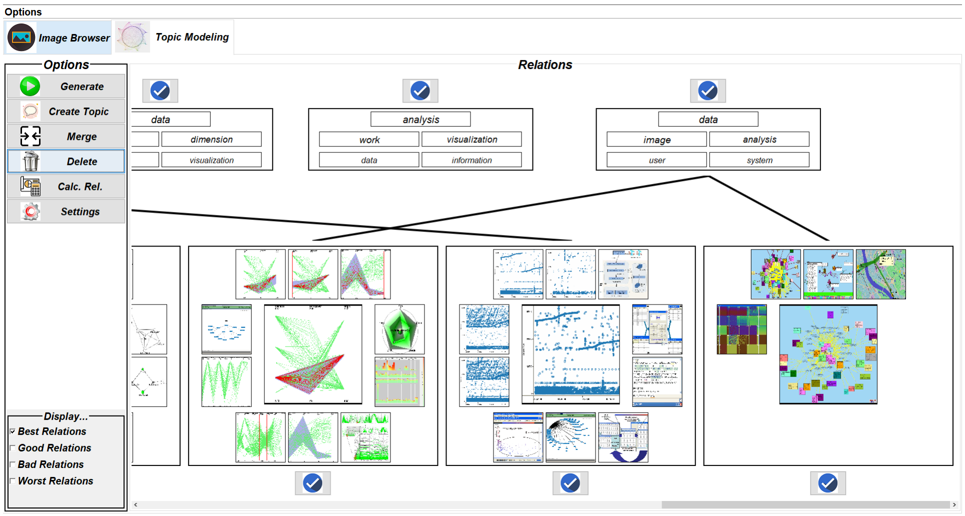
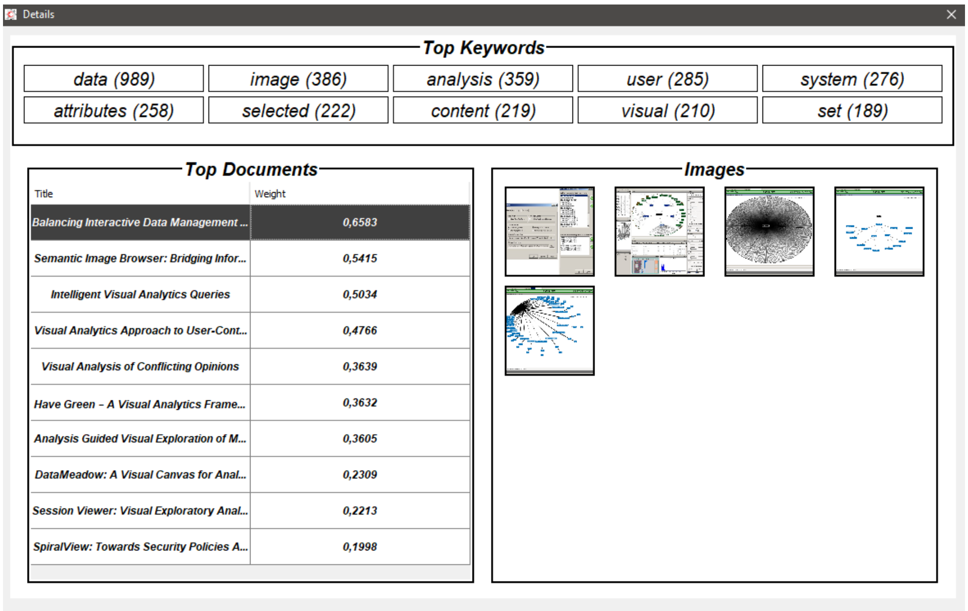


Figure 5.11: Merged Topic with two very good relations

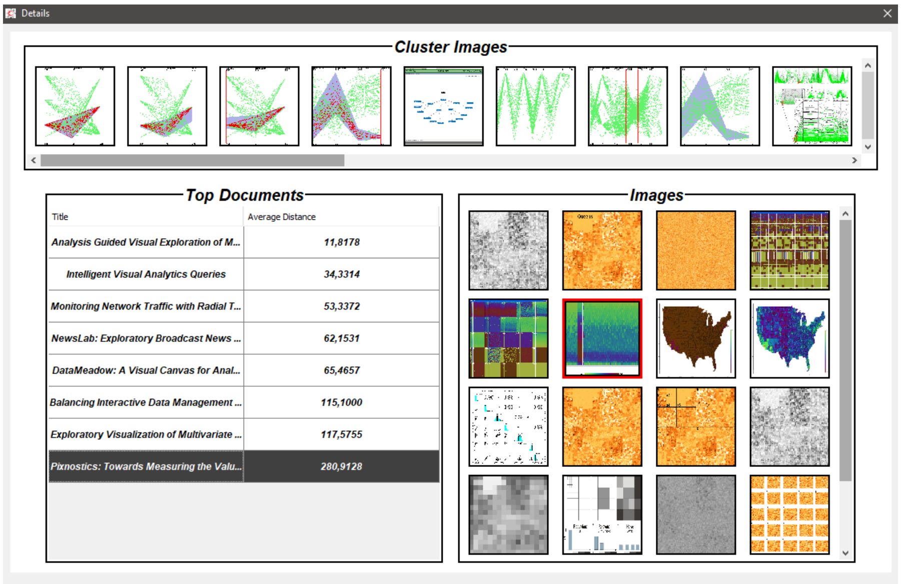


data (989)	image (386)	analysis (359)	user (285)	system (276)
attributes (258)	selected (222)	content (219)	visual (210)	set (189)

Title	Weight
Balancing Interactive Data Management ...	0,6583
Semantic Image Browser: Bridging Infor...	0,5415
Intelligent Visual Analytics Queries	0,5034
Visual Analytics Approach to User-Cont...	0,4766
Visual Analysis of Conflicting Opinions	0,3639
Have Green – A Visual Analytics Frame...	0,3632
Analysis Guided Visual Exploration of M...	0,3605
DataMeadow: A Visual Canvas for Anal...	0,2309
Session Viewer: Visual Exploratory Anal...	0,2213
SpiralView: Towards Security Policies A...	0,1998

Figure 5.12: Details of the merged topic

Then we investigated one of the two connected image clusters and looked at their most relevant documents. It turned out that one of the documents (Pixnostics: Towards Measuring the Value of Visualization, [SSK06]) was indeed a paper (shown in 5.13) which fits the outgoing topic pretty well. It talks about Pixnostics, a technique that automatically analyses pixel images and ranks them according to the potential value for a user which enables the user to obtain insights into data much faster and allows a more effective and efficient visual data analysis process. From this result we can assume that it is possible to find better results by merging them together. Especially if some topics don't even have good relations and in this particular situation seem worthless, they are still useful for combining with other topics to may generate a new topic with a broader spectre of the including cluster's textual contents.



Title	Average Distance
Analysis Guided Visual Exploration of M...	11,8178
Intelligent Visual Analytics Queries	34,3314
Monitoring Network Traffic with Radial T...	53,3372
NewsLab: Exploratory Broadcast News ...	62,1531
DataMeadow: A Visual Canvas for Anal...	65,4657
Balancing Interactive Data Management ...	115,1000
Exploratory Visualization of Multivariate ...	117,5755
Pixnostics: Towards Measuring the Valu...	280,9128

Figure 5.13: New found topic related document inside connected image cluster

5.4 Usecase: Relations between Topics and Image Clusters

5.4.1 Description

In simple words we can say that image clustering is grouping similar image items together. If we look at the text domain, topic modeling is popular and fairly successful. For getting more insights and to obtain a better understanding of the overall document collection, the exploration system gives the user the opportunity to combine those two processes by introducing weighted relations that connect image clusters with topic models. This should not only provide new image cluster depending findings in the context of topic related documents but also provide a kind of assignment of a number of clusters to topics.

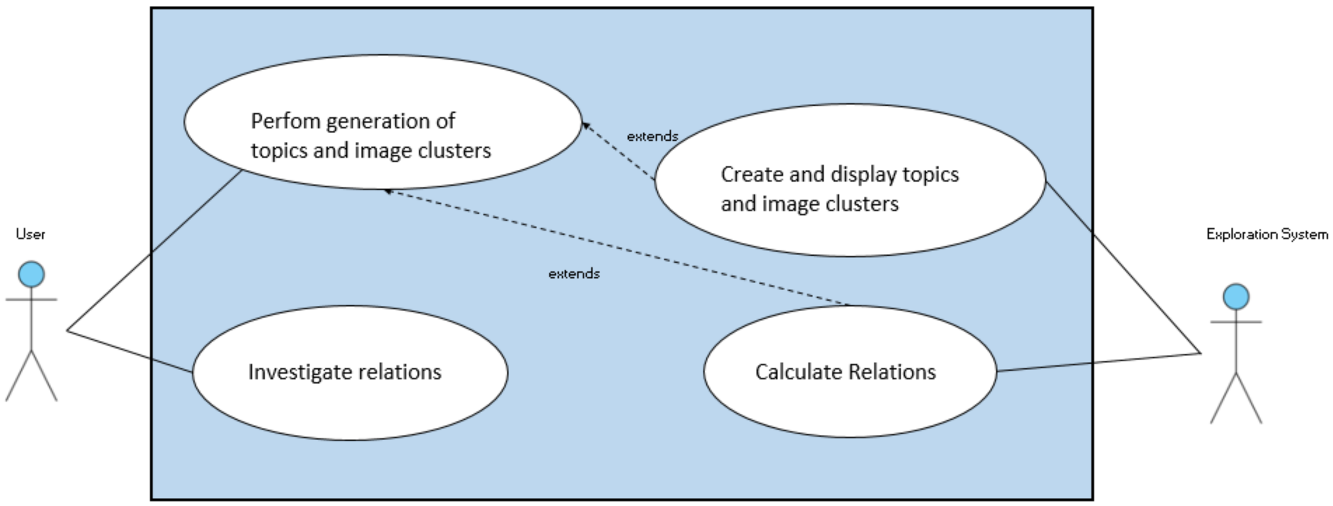

Perfom generation of topics and image clusters
extends
Create and display topics and image clusters
User
extends
Exploration System
Investigate relations
Calculate Relations

Figure 5.14: Use Case Diagram: Calculate Relations between topics and image clusters

5.4.2 Main Scenario

1. Extract or Import Data from a document collection
2. User opens Topic Model Tab
3. User sets number of Topics/Clusters in Settings
4. User presses “Generate” Button
5. Systems displays window with optional Filter Options (for used documents in the modelling process)
6. User states some filters (if needed) and presses “Start”
7. System generates and displays topics (boxes with the 5 most important keywords inside the topic) and image clusters (boxes with a center image surrounded by its most similar images)
8. User presses “Calc. Rel.” Button
9. System calculates relations, connects the topics/clusters by drawing a line. The color of the line represents the weight of the relation
10. User can choose which relations will be displayed

11. User right-clicks on Topic/Cluster

12. The system opens Popup Menu

13. User selects “Inspect Relations”

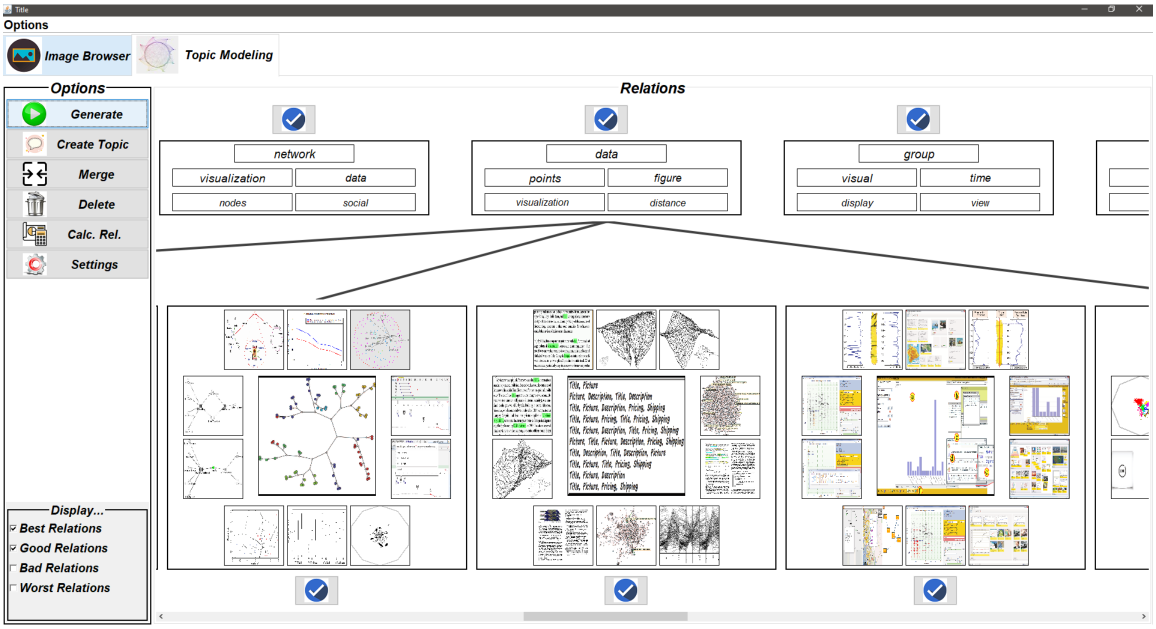
14. System opens a window with all relations of the Topic/Cluster and their corresponding weights

15. User select one relation of his own choice

16. The system opens a tab where it provides data about the relation (involved documents, according cluster and topic, images, weights, etc.)

5.4.3 Finding

This finding deals with the question of how can we find relations between topics and image clusters and does that help us to find new topic related documents. Let’s begin with a closer look to an example of a topic model shown in figure 5.15. We can see that the five most relevant keywords of the central positioned topic model are: data, points, figure, visualization and distance. We also identify three very good relations between the investigated topic and the different generated image clusters. After a close lookup onto those relations (shown in figure 5.16) we can recognize that all matching docs of the best relation are talking about the visualization and analysis of multi-dimensional data. Lets look onto some text snippets out of their abstract section:



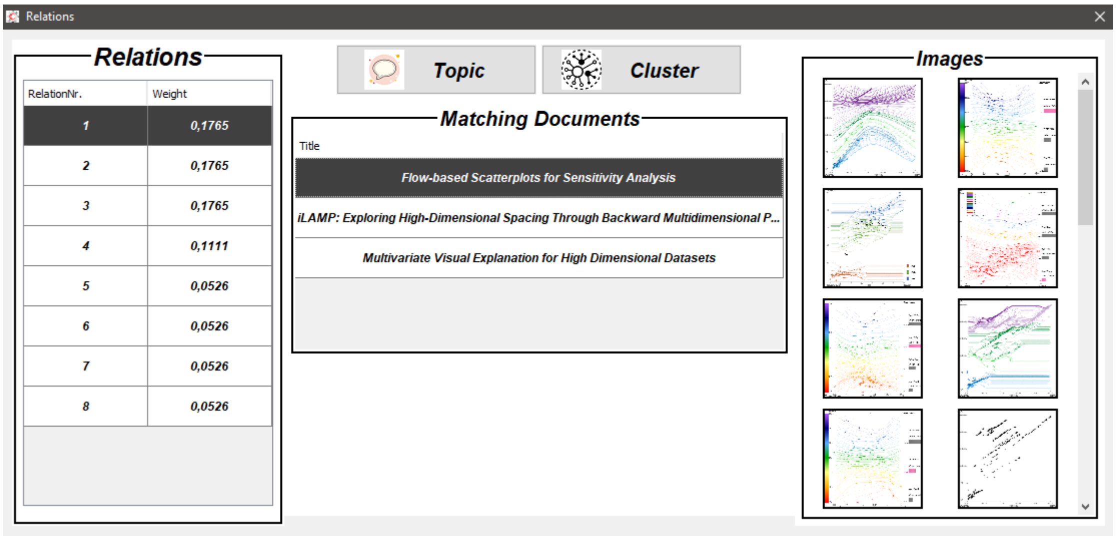
☑ Best Relations
☑ Good Relations
□ Bad Relations
□ Worst Relations

Figure 5.15: Three good relations between topic and clusters

1. **Flow-based Scatterplots for Sensitivity Analysis** [CCM10]
...Visualization of multi-dimensional data is challenging due to the number of complex correlations that may be present in the data but that are difficult to be visually identified. One of the main causes for this problem is the inherent loss of information that occurs when high-dimensional data is projected into 2D or 3D...

2. **iLAMP: Exploring High-Dimensional Spacing through Backward Multidimensional Projection** [dBD*12]
   …We present an inverse linear affine multidimensional projection, coined iLAMP, that enables a novel interactive exploration technique for multidimensional data. iLAMP operates in reverse to traditional projection methods by mapping low-dimensional information into a highdimensional space…

3. **Multivariate Visual Explanation for High Dimensional Datasets** [BTY*08]
   …Understanding multivariate relationships is an important task in multivariate data analysis. Unfortunately, existing multivariate visualization systems lose effectiveness when analyzing relationships among variables that span more than a few dimensions…



Relations

RelationNr.	Weight
1	0,1765
2	0,1765
3	0,1765
4	0,1111
5	0,0526
6	0,0526
7	0,0526
8	0,0526

Matching Documents

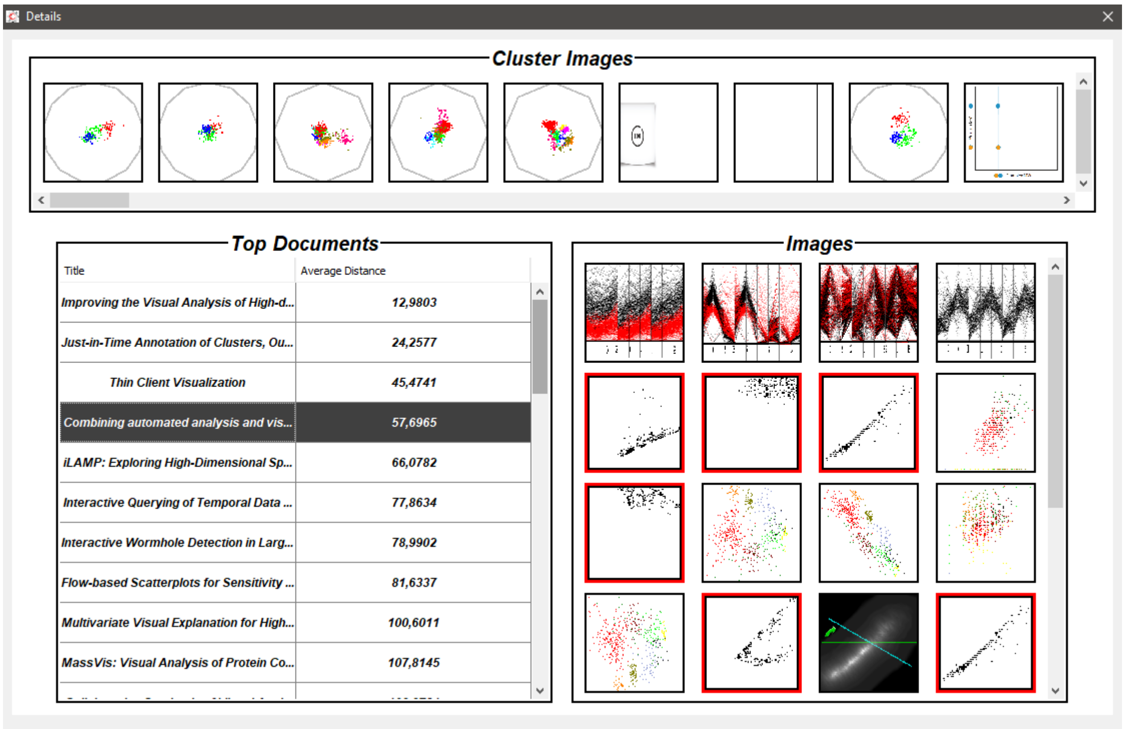
Title
Flow-based Scatterplots for Sensitivity Analysis
iLAMP: Exploring High-Dimensional Spacing Through Backward Multidimensional P...
Multivariate Visual Explanation for High Dimensional Datasets

Figure 5.16: Matching documents of the best relation

The goal now was to find documents within the image clusters which have nearly the same topic than the most relevant papers of the start topic. And indeed we were able to actually find some specific papers which haven't even been stated in the list of relevant documents of the start topic, and we only found them by calculating and investigating the outgoing relations to a specific image cluster (shown in figure 5.17). Let's take a look at them, again with some text snippets taken out of the abstract section and as we can see they don't differ too much from the starting topic:

1. **Combining automated analysis and visualization techniques for effective exploration of high-dimensional data** [TAE*09]
   …Visual exploration of multivariate data typically requires projection onto lower-dimensional representations. The number of possible representations grows rapidly with the number of dimensions, and manual exploration quickly becomes ineffective or even unfeasible…

2. **Improving the Visual Analysis of High-dimensional Datasets Using Quality Measures** [AEL*10]
   …Modern visualization methods are needed to cope with very high-dimensional data. Efficient visual analytical techniques are required to extract the information content in these data. The large number of possible projections for each method, which usually grow quadrat-ically or even exponentially with the number of dimensions…



Title	Average Distance
Improving the Visual Analysis of High-d...	12,9803
Just-in-Time Annotation of Clusters, Ou...	24,2577
Thin Client Visualization	45,4741
Combining automated analysis and vis...	57,6965
iLAMP: Exploring High-Dimensional Sp...	66,0782
Interactive Querying of Temporal Data ...	77,8634
Interactive Wormhole Detection in Larg...	78,9902
Flow-based Scatterplots for Sensitivity ...	81,6337
Multivariate Visual Explanation for High...	100,6011
MassVis: Visual Analysis of Protein Co...	107,8145

Figure 5.17: New found topic related paper taken out from the connected image cluster

We can assume that it is possible to find topic related documents without looking only on corresponding textual information but also at generated image clusters that focus on image similarity alone.

5.5 Usecase: Color-Clustering

5.5.1 Description

In real-world examples an user of a document exploration system often deals with the problem of not exactly knowing the content of a document collection unless the user doesn't face it for the first time. Hence, for the user, several questions appear: What are the general topics of the documents, which images are involved, when were the documents published, what authors do they have and many others. To get a first overview about the document collection the Image Browser of this exploration system was implemented. The user can see all extracted images of the overall document collection and can also investigate the corresponding documents and their meta-data (keywords, authors, title, publisher date, ...). A special feature in this investigation process is the marking of images due to their corresponding meta-data. This results in a kind of color clustering which gives the user a good opportunity to get a good first impression over the dealt document collection.

5.5.2 Main Scenario: Mark images according to selected meta data

1. Extract or Import Data from a document collection

2. User opens Image Browser

3. System shows all images and their most similar images

4. User selects meta data of an image of his choice

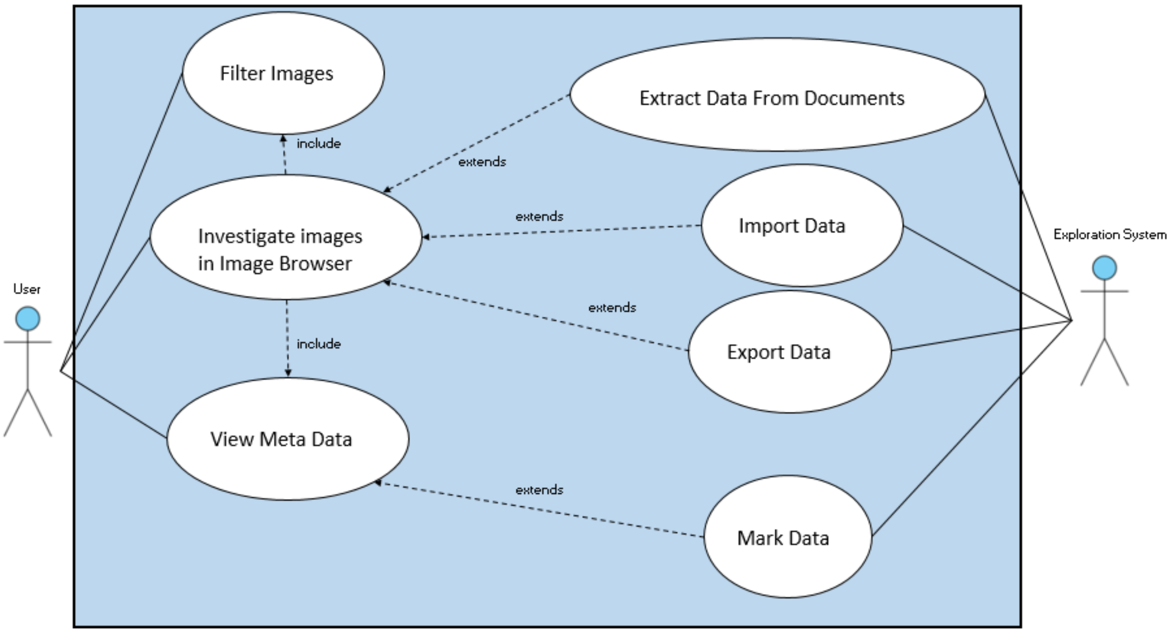


Figure 5.18: Use Case Diagram: Color Clustering

5. System shows meta data

6. User marks meta data of his choice with a color

7. System marks every image in the image browser with the same color which has the same meta data.

Steps 4-7 can be repeated with different images and different meta-data. The user can identify images which fit more than one meta-data by looking at their number of marks.

5.5.3 Alternative Scenario: Color-Clustering with filter options

1. Extract or Import Data from a document collection

2. User opens Image Browser

3. User opens Settings and changes the number of displayed similar Images (for example only 10)

4. System shows all images and only a certain number of similar images (previously defined from the user to focus only on the best results)

5. User investigates those images by looking at their meta data

6. User marks meta data of his choice with a color

7. System marks every image in image browser with the same color which has the same meta data

8. User clicks on filter

9. System opens a window where the user can define filters based on the meta data of the images

10. User applies different filters

11. System displays only Base Images (and their similar images) which are fitting the filter criteria

5.6 Usecase: Document-Browsing

5.6.1 Description

To get a first overview about the document collection the Document Browser displays all documents as Document Cards. The user gets a first impression by viewing onto some meta-data, collections of images and a word cloud which serves as an approximation for the document content. Additionally the the user can expand the exploration for each individual document by switching to the image browser.

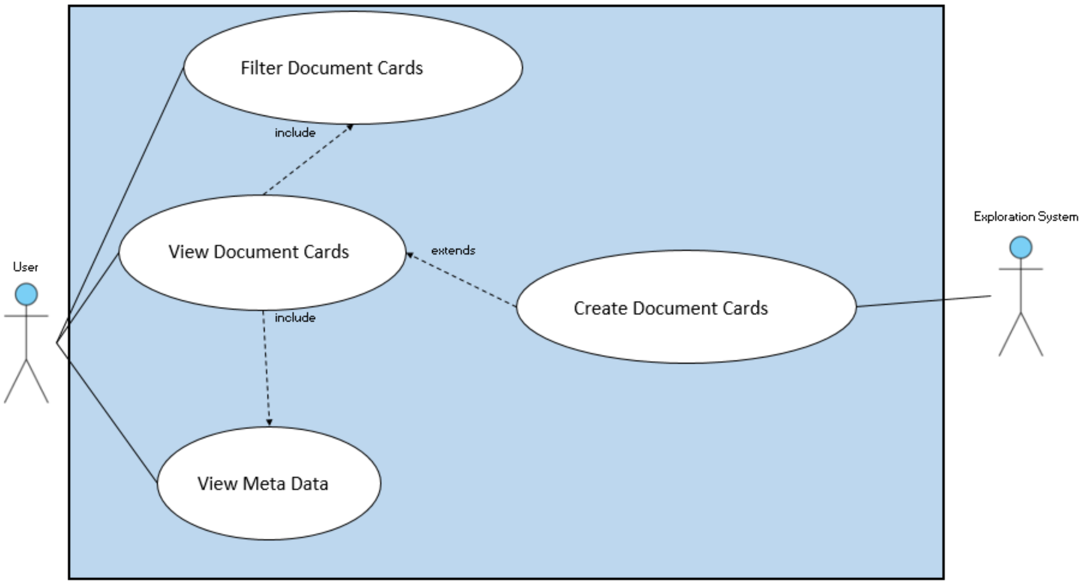


Figure 5.19: Use Case Diagram: Document-Browsing

5.6.2 Main Scenario: Investigate Documents

1. Extract or Import Data from a document collection
2. User opens Document Browser
3. System shows all extracted documents as document cards
4. User sets a filter (e.g. publisher date ”2008”)
5. System only displays document cards that are published in ”2008”
6. User selects ”Show Meta-data” by right-clicking onto document card of his own choice
7. User selects ”Investigate images” by right-clicking onto the same document card
8. System opens Image Browser and shows the document’s images
9. User can continue investigating images, explained in use case 5.5

Chapter 6

Discussion

In the previous section we have discussed several use cases for visualizing information and gain new insights of our document collection. In this section, we will discuss the limitations of our exploration system and try to state possible improvements.

6.1 Extracting document content with Apache PDFBox

The java library gives us the opportunity to extract text, meta-data and images of PDF files. But it has certain limitations. First we can't identify whether the extracted text is coming from table content or figure descriptions. We just get one big string as result and have to deal with that. Improvements would be additional pre-processing methods to actually distinguish between normal text, table content and figure descriptions. Additionally the library really depends on how the PDF files were created because often it is not capable of extracting all available meta-data (for example missing authors, titles,...). Sometimes the extraction of normal text itself can also trigger problems: Since some PDF documents are just images that have been scanned in, the library only returns us images of the whole pages. The text content on pages could also be read in different orders because the cursor could move differently.

6.2 K-Mean Clustering Algorithm

The image clustering via K-mean is relatively simple to implement and it scales the large data sets relatively good. It guarantees convergence and easily adapts to new images. On the other hand the random choice of centroids in the beginning of the algorithm and specific image outliers have a huge impact on the overall result of the clusters. Centroid images can be dragged by outliers or they form a own cluster instead of beeing ignored by the algorithm. Therefore to actually improve the results it would be better to remove certain outliers before starting the clustering process. Since the K-mean is a unsupervised method too we are also not able to verify results which complicates assessing quality. But one of the main problems of the K-mean is choosing K manually. To overcome this problem and detect the optimal choice of K the usage of the Elbow Method is suggested, where the intra-cluster variation or total within-cluster sum of square (WCSS) is minimized to provide better results.

6.3 Topic Modeling with the Latent Dirichlet Allocation (LDA)

Here we have the same problem as in K-mean clustering since the number of topics must be known beforehand. Further limitations of the Dirichlet topic distribution would be that it cannot capture correlations between topics and it can't model the sentence structure because it assumes words are exchangeable.

Chapter 7

Conclusion and Future Work

Document visualization techniques enables researchers to efficiently browse, explore and understand documents. But before all this can be done the contained data has to be pre-processed to derive the desired text and image features. We heard typical steps like stopword removal (most frequent words in a language), stemming (porter stemmer) and other useful approaches. We introduced information retrieval and some of its design principles. We discussed visualization, including its definition and the difference between several common information visualization techniques. We then talked about our own exploration system and presented the developed methods focusing on the implementation, design, strengths, limitations and results. In the following we will outline some recommendations based on the project's challenges and the most promising directions for future research.

Existing visualization methods can be extended to make them more suitable for large document collections. Especially if we think of the possibility of combing and optimizing different methods to achieve a more effective performance and to provide mature results of document information. In our exploration system we combined topic modeling with image clusters. Another possibilities would be TopicNet [GOB*12], which is a combination of graph visualization with topic modeling to display topics and their relationships simultaneously, or TIARA [LZP*09] that combines tag clouds and ThemeRiver [HHN00]. Practical projects also often deal with the problem involving a big variety of data types (multidimensional, text, images and many more). The combined usage of different visual representations can solve that problem by giving more different perspectives to users. A very nice example for this is the Doxplorer with RadVizDoc visualization that visualizes a two-dimensional document landscape in which documents are positioned as anchor points within the radviz based on search queries [Eri20]. The same goes for text data mining which is the process of deriving high-quality information from texts. If we apply more detailed visualization methods to the process of text data mining, for example in the production of granular taxonomies, sentiment analysis or document summarization, we can actually increase the accuracy and quality of the results.

If we look onto the evaluation of many information visualization methods we can recognize that they lack of measurements that indicates quality and many other evaluative possibilities. Fortunately there exists the BELIV workshop series, which is an event focusing on research methods in visualization in which discussions are spanned from new and not-yet fully established evaluation methods, to methods that establish the validity and scope of acquired visualization knowledge. [BEL]

Bibliography

[AEL*10] Albuquerque G., Eisemann M., Lehmann D. J., Theisel H., Magnor M.: Improving the visual analysis of high-dimensional datasets using quality measures. In *2010 IEEE Symposium on Visual Analytics Science and Technology* (2010), pp. 19–26.

[BEL] Beliv workshop. https://beliv-workshop.github.io/. Accessed: 2020-09-28.

[BNJ03] Blei D. M., Ng A. Y., Jordan M. I.: Latent dirichlet allocation. *J. Mach. Learn. Res. 3*, null (Mar. 2003), 993–1022.

[BPS17] Bartram L., Patra A., Stone M.: Affective color in visualization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, Association for Computing Machinery, p. 1364–1374.

[BTY*08] Barlowe S., Tianyi Zhang, Yujie Liu, Yang J., Jacobs D.: Multivariate visual explanation for high dimensional datasets. In *2008 IEEE Symposium on Visual Analytics Science and Technology* (2008), pp. 147–154.

[CCM10] Chan Y., Correa C. D., Ma K.: Flow-based scatterplots for sensitivity analysis. In *2010 IEEE Symposium on Visual Analytics Science and Technology* (2010), pp. 43–50.

[CGK*07] Chang R., Ghoniem M., Kosara R., Ribarsky W., Yang J., Suma E., Ziemkiewicz C., Kern D., Sudjianto A.: Wirevis: Visualization of categorical, time-varying data from financial transactions. In *2007 IEEE Symposium on Visual Analytics Science and Technology* (2007), pp. 155–162.

[Cro06] Croft W.: *Combining Approaches to Information Retrieval*, vol. 7. 04 2006, pp. 1–36.

[Cro16] Crockett D.: Direct visualization techniques for the analysis of image data: the slice histogram and the growing entourage plot. *International Journal for Digital Art History*, 2 (Oct. 2016).

[dBD*12] dos Santos Amorim E. P., Brazil E. V., Daniels J., Joia P., Nonato L. G., Sousa M. C.: ilamp: Exploring high-dimensional spacing through backward multidimensional projection. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2012), pp. 53–62.

[EBJ06] Erol B., Berkner K., Joshi S.: Multimedia thumbnails for documents. In *MM '06* (2006).

[Eri20] Eric G.: A visually aided interactive content-based digital exploration system for document collections.

[FPR09] Fashandi H., Peters J., Ramanna S.: L2 norm length-based image similarity measures: Concrescence of image feature histogram distances. *Proceedings of the IASTED International Conference on Signal and Image Processing, SIP 2009* (01 2009), 178–185.

[GD09] Goker A., Davies J.: *Information Retrieval Models*. 2009, pp. 1–19.

[GOB*12] Gretarsson B., O'Donovan J., Bostandjiev S., Höllerer T., Asuncion A., Newman D., Smyth P.: Topicnets: Visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST) 3* (02 2012).

[GZL*14] Gan Q., Zhu M., Li M., Liang T., Cao Y., Zhou B.: Document visualization: An overview of current research. *Wiley Interdisciplinary Reviews: Computational Statistics 6* (01 2014).

[Ham17] Hamid A.: Relevance feedback in information retrieval systems, 10 2017.

[HHN00] Havre S., Hetzler B., Nowell L.: Themeriver: Visualizing theme changes over time. pp. 115 – 123.

[HSWL12] Han P., Shen S., Wang D., Liu Y.: The influence of word normalization in english document clustering. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)* (2012), vol. 2, pp. 116–120.

[HWV09] Ham F., Wattenberg M., Viégas F.: Mapping text with phrase nets. *IEEE transactions on visualization and computer graphics 15* (11 2009), 1169–76.

[JWL*08] Jeong D. H., Wenwen Dou, Lipford H. R., Stukes F., Chang R., Ribarsky W.: Evaluating the relationship between user interaction and financial visual analysis. In *2008 IEEE Symposium on Visual Analytics Science and Technology* (2008), pp. 83–90.

[KdM19] Kulahcioglu T., de Melo G.: Paralinguistic recommendations for affective word clouds. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2019), IUI '19, Association for Computing Machinery, p. 132–143.

[KM02] Kowalski G., Maybury M.:. In *Information Storage and Retrieval Systems Theory and Implementation* (2002), vol. 8.

[Knu97] Knuth D. E.: *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., USA, 1997.

[LDA] java-lda. https://github.com/chen0040/java-lda. Accessed: 2020-09-11.

[LHB*15] Lohmann S., Heimerl F., Bopp F., Burch M., Ertl T.: Concentri cloud: Word cloud visualization for multiple text documents. In *2015 19th International Conference on Information Visualisation* (2015), pp. 114–120.

[lir] Lire: Lucene image retrieval. http://www.lire-project.net/. Accessed: 2020-09-11.

[LSD*10] Lipford H. R., Stukes F., Dou W., Hawkins M. E., Chang R.: Helping users recall their reasoning process. In *2010 IEEE Symposium on Visual Analytics Science and Technology* (2010), pp. 187–194.

[LZP*09] Liu S., Zhou M., Pan S., Qian W., Cai W., Lian X.: Tiara: Interactive, topic-based visual text summarization and analysis. vol. 3, pp. 543–552.

[MRBK17] Mabrouk D., Rady S., Badr N., Khalifa M. E.: A survey on information retrieval systems' modeling using term dependencies and term weighting. In *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)* (2017), pp. 321–328.

[MRS08] Manning C. D., Raghavan P., Schütze H.: *Introduction to information retrieval.* Cambridge University Press, Cambridge, 2008.

[NSNW13] Niwattanakul S., Singthongchai J., Naenudorn E., Wanapu S.: Using of jaccard coefficient for keywords similarity. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2013* (03 2013).

[pdf] The apache software foundation - apache pdfbox. https://pdfbox.apache.org/. Accessed: 2020-09-11.

[PPK12] Praveen Kumar P., Parna D., K.D.Venkata Rao: Compact descriptors for accurate image indexing and retrieval: Fcth and cedd. In *International Journal of Engineering Research & Technology* (2012), vol. 1, IJERT.

[PST] Download porter-stemmer jar file with all dependencies. https://jar-download.com/artifact-search/porter-stemmer. Accessed: 2020-09-14.

[SGG*18] Shao L., Glatz M., Gergely E., Müller M., Munter D., Papst S., Schreck T.: Extending Document Exploration with Image Retrieval: Concept and First Results. In *EuroVis 2018 - Posters* (2018), Puig A., Raidou R., (Eds.), The Eurographics Association.

[SHD] Damon crockett research projects gallery. http://damoncrockett.com/projects/slicehist.htm. Accessed: 2020-09-28.

[SJW97] Sparck Jones K., Willett P. (Eds.): *Readings in Information Retrieval.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[SOR*09a] Strobelt H., Oelke D., Rohrdantz C., Stoffel A., Keim D. A., Deussen O.: Document cards: A top trumps visualization for documents. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1145–1152.

[SOR*09b] Strobelt H., Oelke D., Rohrdantz C., Stoffel A., Keim D. A., Deussen O.: Document cards: A top trumps visualization for documents. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1145–1152.

[SSK06] Schneidewind J., Sips M., Keim D. A.: Pixnostics: Towards measuring the value of visualization. In *2006 IEEE Symposium On Visual Analytics Science And Technology* (2006), pp. 199–206.

[SWS*00] Smeulders A. W. M., Worring M., Santini S., Gupta A., Jain R.: Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 12 (2000), 1349–1380.

[TAE*09] Tatu A., Albuquerque G., Eisemann M., Schneidewind J., Theisel H., Magnork M., Keim D.: Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *2009 IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 59–66.

[TL04] Tsikrika T., Lalmas M.: Combining evidence for web retrieval using the inference network model: An experimental study. *Information Processing & Management 40* (04 2004), 751–772.

[TS06] Thammasut D., Sornil O.: A graph-based information retrieval system. In *2006 International Symposium on Communications and Information Technologies* (2006), pp. 743–748.

[XHLL16] Xiong C., Hua Z., Lv K., Li X.: An improved k-means text clustering algorithm by optimizing initial cluster centers. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)* (2016), pp. 265–268.

[xst] Xstream. http://x-stream.github.io/. Accessed: 2020-09-11.

[Zha08] Zhang J.: *Visualization for Information Retrieval*, vol. 23. 01 2008.

List of Figures