

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Masterarbeit Nr. 119

**Erweiterung und Evaluation einer
Iubenbasierten Technik
zur Exploration von Textsammlungen**

Ivan Assenov

Studiengang: Informatik

Prüfer: Prof. Dr. Thomas Ertl

Betreuer: Qi Han, Dr. Steffen Koch, Markus John

begonnen am: 25.7.2016

beendet am: 24.1.2017

CR-Klassifikation: H.5.2, I.2.7

Abstract

In recent years there has been a sharp increase in the amount of text publicly accessible in digital form. The primary cause for this is widespread access to the Internet, the popularity of e-mail and social networking websites and collaborative efforts to preserve and share knowledge. These developments have inspired the creation of a wide variety of information visualization techniques that focus on large-scale text data and facilitate its exploration and analysis. One popular approach represents individual documents as glyphs on a 2D surface, with pairwise distances corresponding to semantic similarities. The metaphor of a moveable lens that summarizes the contents of texts underneath it has been proposed as a method of interaction targeted at free exploration tasks. The main goal of this master's thesis project is to extend the basic technique by adding labels to the visualization that guide its users towards regions of interest more quickly without negatively impacting the lens' usefulness. Also, an automatic framework that determines the tool's effectiveness under different parameter settings is developed. Finally, the proposed improvements and the overall technique are evaluated by means of a think-aloud user study.

Zusammenfassung

In den letzten Jahren hat es ein deutliches Wachstum an öffentlich zugänglichen Texten in Digitalform gegeben. Der Hauptgrund dafür ist der Zugriff auf das Internet, die gestiegene Popularität von E-Mails, soziale Netzwerke, und der kollaborative Versuch von Speicherung und Verbreitung von Wissen. Durch diese Entwicklung sind viele Informationsvisualisierungstechniken entstanden, die es ermöglichen große Datenmenge zu explorieren und analysieren. Ein sehr populärerer Ansatz um Dokumente in einer 2D-Ebene darzustellen sind Glyphen. Dabei entspricht die paarweise Distanz der semantischen Ähnlichkeit. Die Metapher einer verschiebbaren Lupe, die den Inhalt fokussierter Texte zusammenfasst, eignet sich für die freie Exploration in einer 2D-Ebene. Das Hauptziel dieser Masterarbeit war es eine vorhandene Grundtechnik zu erweitern, so dass wichtige Begriffe dargestellt werden die den Benutzer unterstützen interessante Bereiche schneller zu finden. Dabei musste darauf geachtet werden, dass die Erweiterung keine negative Auswirkung auf die vorhandene Analysetechnik hat. Des Weiteren, wurde ein automatisches Framework entwickelt, dass die Effektivität mit verschiedenen Parametern misst. Abschließend wurde eine Benutzerstudie durchgeführt, um die Erweiterungen zu evaluieren.

Table of Contents

List of Figures.....	IX
Chapter 1: Introduction.....	1
1.1 Motivating Examples.....	2
1.2 Aim of This Master's Thesis.....	3
Chapter 2: Background and Related Work.....	6
2.1 Standard Information Visualization Model.....	6
2.2 Natural Language Processing.....	7
2.2.1 Tokenization.....	8
2.2.2 Stemming, Lemmatization and Stopwords.....	8
2.2.3 Bag-of-Words Model.....	10
2.3 Text Visualization Techniques.....	11
2.3.1 Word Clouds.....	12
2.3.2 Document Spatialization.....	14
2.3.3 Focus+Context Techniques.....	16
2.3.4 DocuCompass.....	19
Chapter 3: Project Architecture.....	22
3.1 Existing Implementation.....	22
3.2 System Overview and Technologies.....	25
3.2.1 Prefuse Toolkit.....	25
3.2.2 Application Architecture.....	26
Chapter 4: Implementation.....	29
4.1 Automated Framework.....	29
4.1.1 Automated Logging.....	29
4.1.2 Size of Logging Database.....	31
4.1.3 Document Coverage.....	32
4.1.4 Document Prominence.....	34
4.2 Static Labeling.....	38
4.2.1 Document Clustering.....	39
4.2.2 Non-Occluding Static Labeling.....	42
4.2.3 Static Labeling with Occlusion.....	47
Chapter 5: User Study.....	51
5.1 Research Hypotheses.....	51
5.2 Structure of the Study.....	52
5.2.1 User Logging.....	52
5.2.2 Protocol for Conducting the User Study.....	53
5.3 Result Analysis.....	55
Chapter 6: Conclusion.....	59
Appendix A: User Study Consent Form.....	60
Appendix B: Basic Technique Questionnaire.....	61

Appendix C: Advanced Technique Questionnaire.....	66
Bibliography.....	72

List of Figures

Figure 1.1: Examples of the growth of textual data over time.....	3
Figure 1.2: The DocuCompass TextVis approach.....	4
Figure 2.1: The standard InfoVis reference model.....	7
Figure 2.2: Examples of stemming and lemmatization.....	9
Figure 2.3: Word cloud visualization.....	13
Figure 2.4: The SPIRE TextVis system.....	15
Figure 2.5: Fisheye view of graphs and maps.....	17
Figure 2.6: The document lens technique.....	18
Figure 3.1: The LensMania TextVis technique.....	23
Figure 3.2: Overview of the Prefuse visualization toolkit.....	26
Figure 3.3: LensMania's application architecture.....	27
Figure 4.1: Automated Logging Framework.....	30
Figure 4.2: LensMania's application architecture extended with logging.....	31
Figure 4.3: Document coverage model.....	32
Figure 4.4: Actual document coverage and relative error.....	33
Figure 4.5: Basic prominence model: document view.....	36
Figure 4.6: Basic prominence model: average view.....	37
Figure 4.7: Basic prominence model: best view.....	37
Figure 4.8: Basic prominence model: fixed view.....	38
Figure 4.9: Density-based clustering.....	41
Figure 4.10: First static labeling prototype.....	44
Figure 4.11: Second static labeling prototype.....	46
Figure 4.12: The final static labeling prototype.....	47
Figure 4.13: Visualization of the push-pull algorithm.....	49
Figure 5.1: Document coverage during the user study.....	58

List of Abbreviations

InfoVis	information visualization
TextVis	text visualization
NLP	natural language processing
tf	term frequency
tf-idf	term frequency–inverse document frequency
t-SNE	t-distributed stochastic neighbor embedding
PCA	principle component analysis
docupoint(s)	document point(s)

Chapter 1: Introduction

Throughout history, written text has been seen as a defining characteristic of human civilization. The symbolic representation of natural languages is an often complex but effective system of preserving and conveying information over space and time. However, the massive amount of textual data produced by everyday human enterprise has surpassed our limited capacity to process it in a timely manner. The widespread access to home computers and smartphones and the popularity of e-mail and social networks has made it increasingly difficult to cope with the sheer volume of written text that constantly vies for our attention.

The advent of computers has also made possible the development of techniques to more efficiently sieve through large data, including text. However, employed on their own, even highly advanced approaches from the fields of information retrieval, data mining and computational linguistics usually suffer from two significant drawbacks:

- using them requires *a priori* in-depth knowledge of the contents of a textual corpus;
- they are suitable only for solving problems with well-defined information needs, where the goal of an exploration is specific and known in advance.

Consequently, these methodologies are not readily applicable to the more general task of exploring a completely unfamiliar textual corpus, browsing through its contents for insights rather than searching them for a specific answer. This is perhaps best exemplified by the search engine, the ubiquitous and well-known interface to textual libraries. A search query's results are only meaningful to a

user who knows in advance what their goal is, understands what information they require to solve it and knows how to formulate their data requirement as a query in a natural language (see, e.g., the augmented information retrieval model by Broder [11]). By contrast, a user unfamiliar with the contents of a textual corpus, wishing to broadly explore it, has little use for a search engine.

The research field of information visualization (InfoVis) and its specialized sub-discipline text visualization (TextVis) offer a variety of approaches to transform raw, symbol-based written text into an appropriately chosen visual mapping of its contents. This transformative process may involve (as part of the standard InfoVis reference model proposed by Card et al. [14]) not just the selection of appropriate visual metaphors, but also the use of statistical and linguistic transformations on the text input. The end goal is to facilitate both pre-defined and exploratory tasks and to support sense- and decision-making by taking advantage of the innate pattern recognition abilities of the human brain.

1.1 Motivating Examples

Regardless of the goals of exploration, TextVis techniques are most obviously useful when the sheer size of a textual corpus far exceeds a person’s ability to sieve through even summaries of the documents’ contents. As with other types of information, exactly what “large” textual data means is open to some interpretation.

For an illustrative example, as of January 2017 the volunteer-driven Project Gutenberg, which digitizes and preserves works with expired copyright protection, contains as many as 53,000 e-books on its website¹, with thousands more added each year². Wikipedia³, the free online encyclopedia editable by anyone

1 <https://www.gutenberg.org/>

2 <http://www.gutenbergnews.org/statistics/>

3 <https://www.wikipedia.org/>

and one of the most referenced sources in the world, hosts over 41 million articles across 293 languages⁴ (and over 5 million in English alone; see Figure 1.1 for historical context). Finally, the Library of Congress, the largest library in the world, contains more than 38 million books and other printed materials, with several thousand added each day⁵. These and similar examples clearly demonstrate the need for adequate computer-based search and exploration methods and may explain the impressive variety of TextVis techniques created in recent years. Nevertheless, efficient and effective visualization of large-scale textual data sets remains an open research challenge.

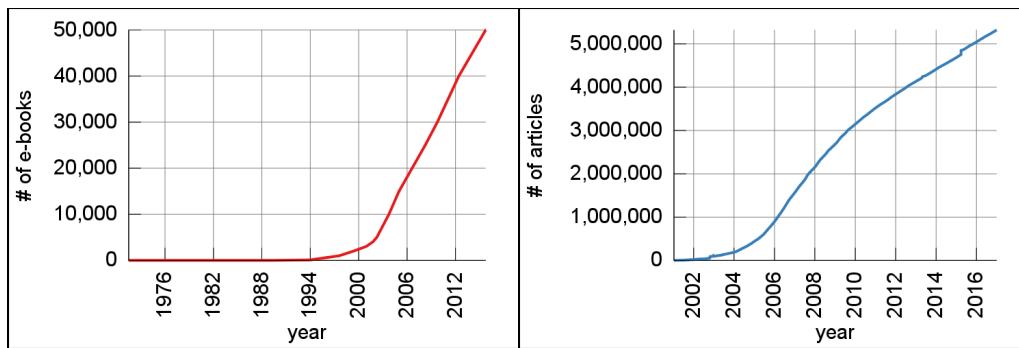


Figure 1.1: Examples of the growth of textual data over time: (left) Project Gutenberg, (right) English Wikipedia. Source: own representation with data from ² and ⁴.

1.2 Aim of This Master's Thesis

A popular approach to visualizing textual corpora is to represent each individual document as a glyph in 2D space. These visualizations, reminiscent of the more general scatterplot InfoVis technique, attempt to map a document's properties to the x - and y -axis of a Cartesian coordinate system, with additional information encoded in an individual data point's shape, size and color. This approach usually provides a good overview of a textual data set, broadly sum-

⁴ https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

⁵ <https://www.loc.gov/about/fascinating-facts/>

marizing its contents. Support for more traditional tasks with well-defined information needs can be added by incorporating into the visualization a search engine or a filter, based on document metadata.

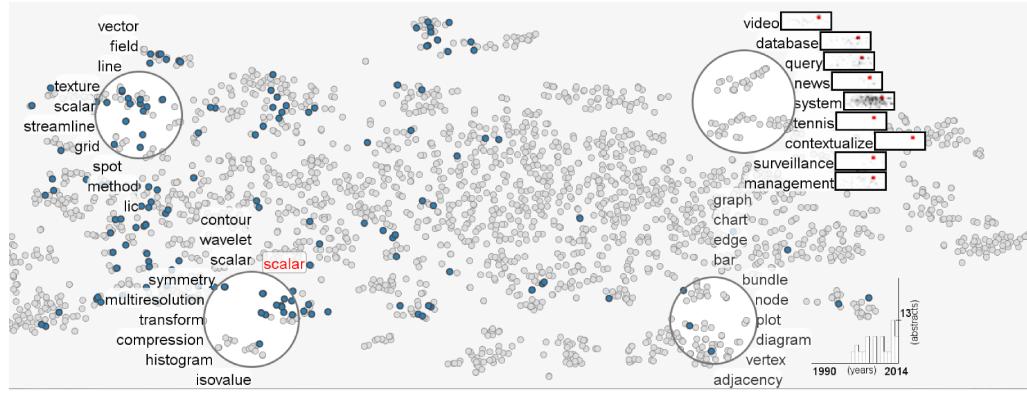


Figure 1.2: The DocuCompass TextVis approach. Individual documents are represented as dots on a 2D grid, with a “magic” lens used for focused exploration. The labels shown around the moving lens are representative of the documents underneath it. Source: [31].

Because the input data set consists of written text, a logical extension to this approach would be to use text in the resulting visualization, e.g., by positioning the document’s title or keywords near its corresponding glyph and by further using font type, size and coloration as visual variables. Unfortunately, this idea does not scale well with the previously discussed vast amounts of textual information created each day, as it would be impossible to individually label millions of data points without substantial overlap and occlusion.

One proposed solution to this challenge is a movable “magic” lens over the 2D representation of the data set, with labeling applied only to the documents directly underneath it. This intuitive interaction supports various granularities of analysis and is well-suited to users wishing to freely explore a textual library at their own pace. DocuCompass [31] (depicted in Figure 1.2), an advanced implementation of the scatterplot and lens techniques, has been found to be an

easy-to-use and versatile tool for exploring textual corpora, with minor deficiencies. One proposed improvement in particular was the addition of static, “always on” labeling to the visualization to more quickly direct the user towards areas of higher interest.

The main goal of this master’s thesis and its accompanying development project is to research different possibilities of adding static labeling to the existing 2D glyph visualization, with which to provide “at-a-glance” summaries of a large document corpus. In cases where this may lead to excessive occlusion of existing visual features or otherwise may detract from the technique’s overall usefulness, proper interaction between the moving lens and the labels is considered. In the process of developing a satisfactory solution, several different approaches to the labeling problem are discussed and prototyped and their pros and cons are assessed.

A secondary goal of this thesis is to develop an automated framework to measure the effectiveness of the DocuCompass approach under different parameter settings like the magic lens’ size or the number of terms shown around it. This may provide insights into the overall usefulness of the technique for exploration tasks and even suggest better placement options for the static labels.

Finally, the most promising static labeling prototype is evaluated with a between-group think-aloud user study with recorded voice and screen captures and logging of user interactions with the tool. Users are presented with both a typical free exploration task and a short goal-oriented one to solve and encouraged to share their opinions of the technique.

Chapter 2: Background and Related Work

The work presented in this master’s thesis builds upon research from various fields of computer science, such as information retrieval and data mining, computational linguistics, as well as InfoVis and its more specialized sub-discipline TextVis. Before discussing the proposed extensions to the DocuCompass approach in detail, a review of existing literature on the subject of visualizing textual corpora is necessary. This chapter serves as an introduction to some of the challenges posed by TextVis and attempts to provide a relevant but non-exhaustive list of some of the most cited approaches.

2.1 Standard Information Visualization Model

Compared to the broader research field of data visualization, TextVis’ defining characteristic is that the raw input data is entirely nominal in nature. Moreover, human languages tend to be highly complex, context-dependent and very redundant by nature, which necessitates a noticeably different methodology when trying to incorporate them into a visualization. While the following subsections discuss these problems and their solutions in greater detail, for the purposes of clarity it may be beneficial to provide a broad overview of the text-to-visualization pipeline here.

Let us consider, for example, the standard InfoVis reference model proposed by Card et al. [14], a variant of which is shown in Figure 2.1. The model depicts the usual sequence of steps required to transform any input data into an interactive visualization that supports a task, whether well-defined or exploratory. Within the context of TextVis, the source data is a textual corpus of any size, but, as discussed in Section 1.1, particularly large data sets pose a greater challenge to human cognitive capabilities and thus increase the demand

for effective visualization techniques. The first computational step is a transformation of this raw data into a representation suitable for further mathematical analysis, e.g., a well-structured table. For nominal textual data, this usually requires the use of natural language processing (NLP) and statistical methods, to be discussed in Section 2.2. The various TextVis techniques presented in Section 2.3 provide examples of the final two steps in this pipeline – the mapping of the tabular structures to visual variables and showing a concrete view of the visualization to the user. It should be noted that this standard model strongly emphasizes the importance of introducing meaningful user interaction with every stage of the process. Furthermore, the choice of algorithms at each step should be made based on the specific information needs of the end user.

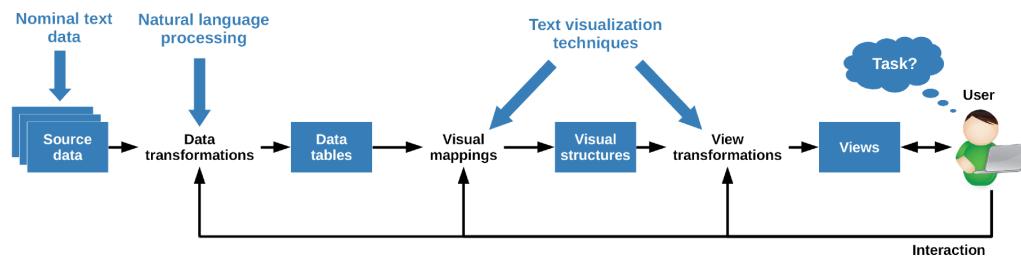


Figure 2.1: The standard InfoVis reference model. The blue arrows indicate the positioning of some of this chapter's topics within it. Source: own representation based on [14].

2.2 Natural Language Processing

Extracting knowledge from the contents of a data set written in a natural human language by, e.g., attempting to summarize its contents, requires the use of data transformation techniques specific to textual data. Many methods from the fields of information retrieval and data mining have been proposed [42], [45]; however, this section of the thesis will focus on those relevant to the DocuCompass approach to TextVis and its extension with new features. Even

so, the following subsections can be considered to present a rather typical NLP pipeline used by a variety of techniques.

2.2.1 Tokenization

Tokenization is the process by which a text corpus is split into its constituent parts (tokens). The granularity of this process depends on the needs of the visualization; a single token in this sense can be a whole document or a separate chapter, paragraph, sentence or word from it. Once determined, the sequence of tokens is pushed down the NLP pipeline for further processing.

Very often, the goal is to split text into its constituent word tokens. This is a rather straightforward task that usually involves one of three approaches: rules-based, dictionary-based or machine learning. In the former, boundaries between individual words are identified by a predefined rule set that is dependent on the targeted human language. Two elementary approaches [21] that often work well are to either isolate strings delimited by whitespace characters (and then remove any remaining punctuation marks from the results) or to identify continuous sequences of alphanumeric characters with embedded hyphens and apostrophes in them⁶. Unfortunately, both of these are only suitable for languages that use a particular character as a word separator; this includes most alphabet-based writing systems, but does not cover languages like Chinese, Japanese, Korean, etc. A dictionary-based approach that depends on a comprehensive lexicon is required in these cases.

2.2.2 Stemming, Lemmatization and Stopwords

After tokenization is complete (and assuming tokens consist of individual words), a number of additional methods can optionally be applied to its output to improve future analysis. Stemming is the process of reducing words derived

⁶ The latter be achieved with an extended regular expression: ([[:alnum:]]|-|')+

from the same morphological root to a common base form, usually by iteratively removing word prefixes and suffixes in a language-dependent manner [37]. Algorithms to achieve this, such as the Porter Stemmer [47], are very efficient and thus applicable even to large textual corpora, because they work directly with the individual tokens and ignore the context the words were used in. Unfortunately, it is often the case that the result of the transformations is itself not a word in the lexicon of the target language, thus making it confusing to the user and unsuitable for direct use in the final visualization. An illustrative example of the Porter Stemmer algorithm is depicted in Figure 2.2 (left).

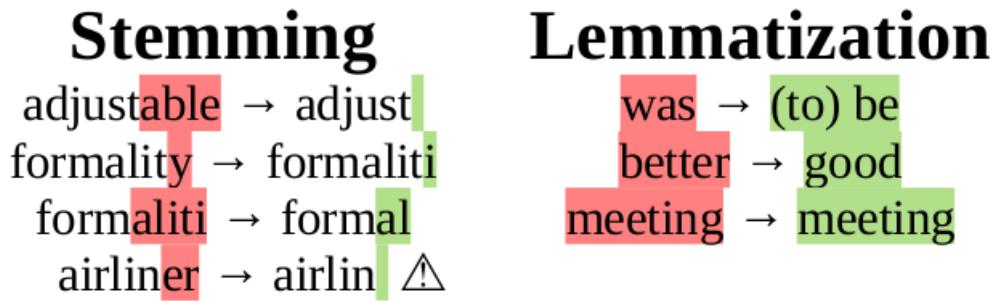


Figure 2.2: Examples of stemming (left) and lemmatization (right). A red coloring indicates the parts of a token considered for replacement, green shows the actual transformations. Worth noting are: (1) how the word “formality” is stemmed iteratively, (2) how the word “airliner” produces the non-word “airlin”, (3) how the word “meeting” is recognized by the lemmatizer as the noun “meeting” and not as the verb “to meet”. Source: own representation using the stemming algorithm from [47].

Lemmatization, on the other hand, aims to identify the lemma – the canonical or dictionary form – of a token by morphological analysis and dictionary lookup. This is significantly more complicated and expensive than stemming, because the algorithm must take into account the context of use of a word, correctly characterize its part-of-speech category and resolve any ambiguities by looking at redundant information encoded in a human language. This may make lemmatization prohibitively slow for real-time visualizations of large-

scale libraries. However, the technique produces grammatically correct tokens, which are understandable to users and can be directly used in a visualization. An example of lemmatization is depicted in Figure 2.2 (right).

Stopword removal is an easy approach to term filtration which, based on a predefined language-dependent list of words, eliminates those that carry no information about the contents of a text, such as pronouns, conjunctions, prepositions and some of the most commonly used words. Lookup tables for many languages are readily available online⁷, while both lemmatization and stopword removal are supported by the Stanford CoreNLP library for the Java Virtual Machine [43].

2.2.3 Bag-of-Words Model

The output of the NLP techniques discussed thus far is a sequence of tokens (typically words reduced to some canonical form). This data structure already supports some basic sense-making, like searching for the existence of a word or phrase in a text corpus. It is, however, usually insufficient for more advanced exploration and visualization tasks, in particular summarizing documents' contents and comparing them to one another.

The bag-of-words or vector space model is a common way of representing documents as vectors in vocabulary space. Information about the ordering and context of use of tokens is discarded and replaced by a vector that preserves only a word's weighted multiplicity. Each document of a textual corpus is thus represented as a single vector of key-value (token-frequency) pairs, whose dimensionality is the vocabulary of the entire data set [42]. Different weighing schemes can be applied to the frequency component of a term [17], [51], with the two most popular being:

⁷ <http://members.unine.ch/jacques.savoy/clef/>

- **term frequency (*tf*):** the raw number of occurrences of a term in a document; the resulting vector may be normalized so as to compensate for particularly long or short texts;
- **term frequency-inverse document frequency (*tf-idf*):** $tf \times \log\left(\frac{N}{m}\right)$, where N is the total number of documents in the data set and m is a count of documents in which the term appears; the resulting vector may also be normalized afterward. Incorporating a measure of how often a word appears in the corpus as a whole assigns less significance to terms that are too prevalent and do not properly differentiate documents from one another.

The goal of these transformations is to mathematically model the contents of a collection of documents as vectors that can be compared to one another using, e.g., the Euclidean distance $\|\vec{u} - \vec{v}\| = \sqrt{\sum_{i \in \text{vocabulary}} (u_i - v_i)^2}$ or cosine similarity $\cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|}$, or projected into 2D or 3D space with a dimensionality reduction technique.

2.3 Text Visualization Techniques

The NLP approaches discussed above serve to prepare a textual corpus for further exploration by transforming documents into a vector-based summary of their contents, making them easier to manipulate programmatically. Looking back at the previously discussed standard InfoVis reference model [14] depicted in Figure 2.1, this corresponds to the first step of the processing pipeline where raw input data is transformed into a well-structured tabular representation. This section deals with the latter two stages, namely:

- the mapping of a text corpus' metadata and bag-of-words vectors to visual variables and metaphors that support a specific exploration task;
- presenting a useful view of the visualization to the user, with appropriate interactions that facilitate the sense- and decision-making process.

For the reasons discussed in Section 1.1, there is a growing body of literature, techniques and tools that deal with TextVis. In order to provide proper background and context for DocuCompass' approach to the presented challenges and position it within its scientific field, a brief overview of some existing methodologies is necessary, particularly those employed by the application presented in Section 3.1.

2.3.1 Word Clouds

Word clouds are a classic TextVis technique that predates the advent of digital computers and the formal establishment of the InfoVis field [57]. The basic idea is quite straightforward: after the NLP stage is complete and structured data is available, individual terms are drawn with varying font sizes in proportion to their weighed frequencies of occurrence from the bag-of-words model. In this way, the corpus' more prevalent words appear larger and draw the user's attention to themselves. Besides this simplistic mapping to the visual variable of size, there exist many variations of the approach that produce aesthetically different results depending on how term positioning, direction, colors and font type are chosen, whether or not similar terms are grouped together [13], [61] or clustered [15], how boundaries between documents are depicted [12], [36] and what interactions are supported. Wordle [58] is a popular web-based tool to generate word clouds, while the Word Cloud Explorer [33] offers fine-grained control of every step of the visualization pipeline. The parallel tag clouds technique [20] employs several word clouds at once to visualize differences amongst facets of large-scale text corpora.

In terms of their usefulness as a visualization technique, word clouds are most often employed as an entry point to a large collection or to provide an at-a-glance summary of its contents. According to Rivadeneira et al. [49], they may facilitate the following user tasks among others: locating a specific item or confirming its absence, browsing content with no particular item in mind and getting an overview of the most and least prevalent topics. However, there is evidence that the technique's popularity may have more to do with its widespread use as a social signifier for a group than with its actual applicability to data analysis [28].



Figure 2.3: Word cloud visualization. Source: own representation using Wordle [58] and the data set discussed in Section 3.2.2

An example word cloud visualization of the data set discussed in Section 3.2.2 is shown in Figure 2.3. It provides a decent overview of the text corpus, making it immediately obvious what the overall topic and some of the main sub-topics are. With appropriate user interaction added, it can serve as a table of contents and a point of entry to the data set. However, the boundaries between documents are not preserved and most of the words are illegible.

2.3.2 Document Spatialization

Representing data samples as glyphs on a 2D surface or in 3D space is a hallmark technique of InfoVis and one that has been successfully adapted for TextVis. The most straightforward way to map documents to the axes of a Cartesian coordinate system is by taking advantage of inherently positional metadata, such as geo-location information now widely available due to the proliferation of GPS technology and the popularity of social media. This is the approach to TextVis spatialization employed by, e.g., SensePlace2 [40] and ScatterBlogs [9], [10], both of which facilitate visual analysis of micro-blogging messages by placing labels on a map of the world. Beyond just position, other visual mappings like shape, color and size can be added to a visualization to increase the number of variables shown.

Spatialization based on the contents of a text corpus present more significant challenges; the vectors produced by the bag-of-words model discussed in Section 2.2.3 are very highly dimensional by design, even without the complications of large-scale data. This necessitates the use of dimension reduction techniques that project the vectors in 2D and 3D space, a process that invariably discards information and introduces errors in their pairwise distances. Principle component analysis (PCA) achieves low-dimensional mappings by transforming the input data to a new coordinate system, whereupon components with the least amount of variance are discarded. More recent approaches like least-squares projection [46] and t-distributed stochastic neighbor embedding (t-SNE) [39] reduce errors and significantly improve visualization quality.

TextVis techniques based on 2D and 3D specializations provide a good overview of a data set, but also have the advantage of emphasizing document similarities. They can serve as an entry point to further exploration of a textual corpus and lend themselves well to combination with other visualization ap-

proaches. When dealing with large-scale libraries, the potential pitfall of too much occlusion due to the sheer amount of data points drawn on the surface can be somewhat mitigated by providing limited views of the data and by implementing appropriate interactions.

The SPIRE system described in detail by Wise [59] is one of the earliest, best known and most-cited approaches to TextVis. Beyond its NLP algorithms, it consists of two visualizations: the 2D Galaxies and the 3D ThemeScapes. The former works by reducing the high-dimensional representation of documents to a 2D scatterplot of points whose relative position to one another may reveal similarities, trends and patterns in the text corpus [60]. ThemeScapes extend this approach by providing a fully 3D visualization based around the metaphor of a “text landscape”. Instead of focusing solely on the documents, it uses their physical proximity to determine areas of the plot where a specific word is particularly common and raises the landscape at those positions. Calculated over all the terms in the corpus, the highest elevations occur in places where documents with highly similar contents are present. Implemented interactions with the SPIRE systems enable deeper exploration and even statistical re-evaluation of the corpus. An illustration of the Galaxies and ThemeScapes tools is shown in Figure 2.4.

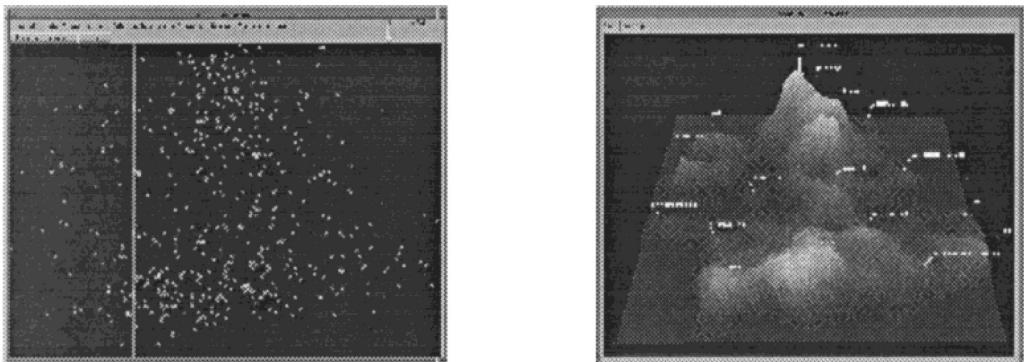


Figure 2.4: The SPIRE TextVis system. Galaxies (left) represents each document as a single star-like point in a 2D scatterplot. ThemeScapes (right) elevates portions of a 3D landscape based on the frequency of occurrence of terms. Source: [59].

A slightly different approach to document spatialization was demonstrated by Chuang et al. [18] with their Stanford Dissertation Browser. In a preprocessing step, the tool analyzes the contents of over 9000 Ph.D. theses by means of either *tf-idf* or latent Dirichlet allocation [8]. The user selects a department at Stanford University to be placed in the center of the 2D plot, with all other departments placed in concentric circles around it based on a measure of cosine similarity between departments. Departments are represented as circles whose sizes depend on the number of documents (theses) associated with them and whose colors indicate the scientific field they belong to.

2.3.3 Focus+Context Techniques

Most visualization scenarios require that users interact with more components than can conveniently be displayed on the screen at one time. Always displaying the entire contents of a textual corpus or even just a summary of them is an unrealistic approach to TextVis, as these will drown out any other visual mappings and the patterns they are supposed to reveal. Therefore, well-designed interactive visualizations aim to first provide a broad overview of an input data set as an entry point to further exploration and view manipulations [53]. When visualizing text documents this usually means showing at any given time only a small, dynamically changing subset of the input text.

According to the classification by Cockburn et al. [19], techniques based around the overview+detail paradigm display both their broad overview and the more detailed views simultaneously, but in physically separated presentation spaces. By contrast, focus+context approaches seamlessly integrate both views into a single display, showing focused objects in full detail without sacrificing any context information. While in practice this is harder to achieve because of the possibility of occlusions of data, the users benefit from not having

to split their attention between different parts of an application’s graphical interface.

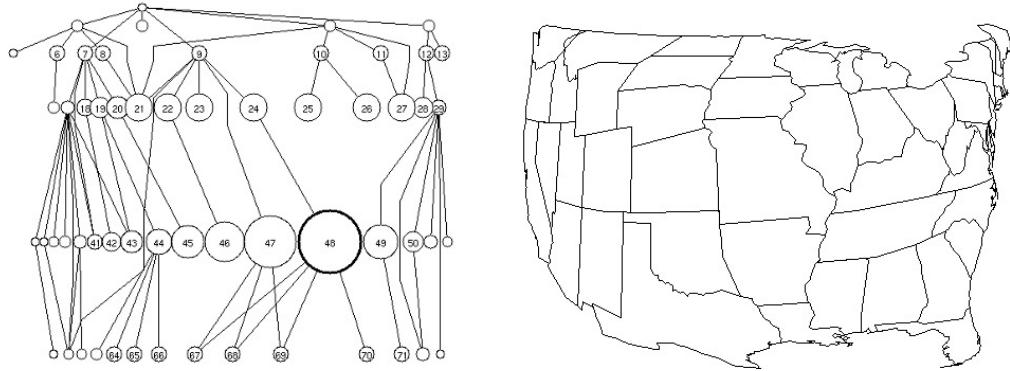


Figure 2.5: Fisheye view of graphs (left) and maps (right). Information that lies away from a point of focus is suppressed based on its Euclidean distance to that point. Source: [52].

Among the many proposed focus+context techniques for varying types of data are polyfocal displays [34], bifocal displays [1] and perspective walls [41]. Many approaches derive from the seminal paper by Furnas [26] (and later revisited in [27]) describing fisheye view interfaces, in which information that lies away from an area of focus is suppressed or diminished by way of defining a “degree of interest” function over the visualization’s contents. Extending the basic idea, Sarkar and Brown [52] presented geometric distortions of graphs and maps based on the Euclidean distance from a focal point (Figure 2.5 depicts their technique). Several other developments of the fisheye concept focus on tree visualizations, such as the Hyperbolic Tree Browser [35] and TreeJuxtaposer [44], and on calendar applications like Table Lens [48] and DateLens [3]. For visualization of text documents, Robertson and Mackinley in 1993 [50] introduced the document lens technique for interaction with page-based representations of text. Using a rectangular lens, the user can focus on a part of the display without losing sight of the global context. Areas outside the focus region

CHAPTER 2: BACKGROUND AND RELATED WORK

are stretched with affine transformations, but kept visible, if not always legible (see Figure 2.6). According to Cockburn et al. [19], fisheye techniques in general suffer from two potential drawbacks caused by the distortion of the information space: misinterpretation of the underlying data and difficulty in acquiring proper targets for focus.

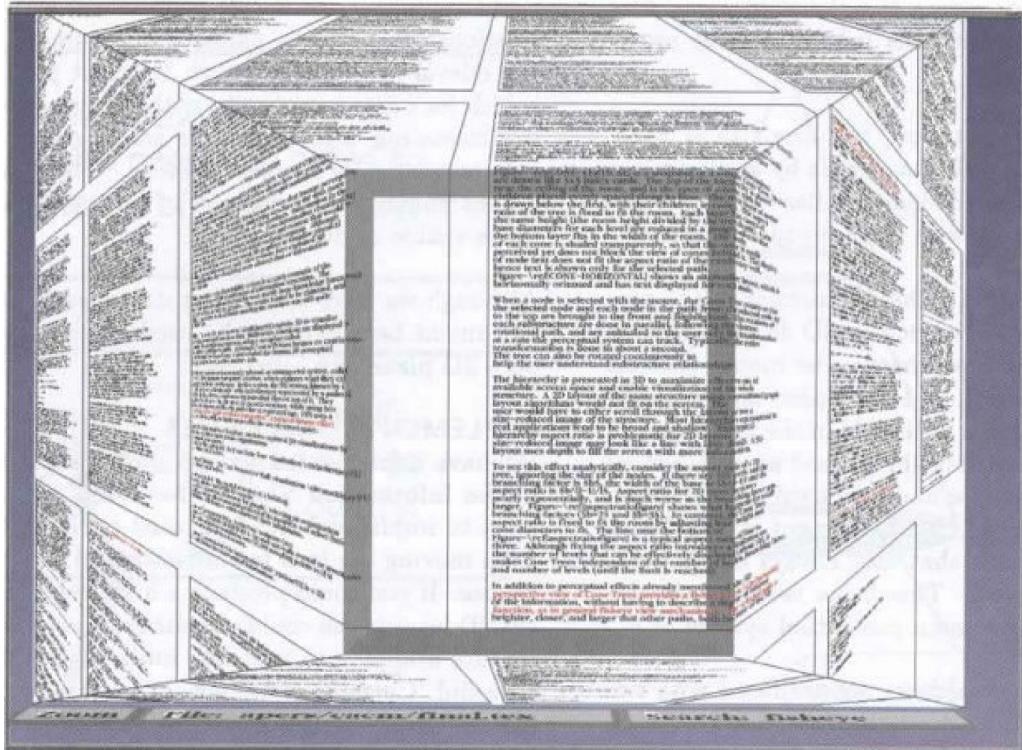


Figure 2.6: The document lens technique. A rectangular, moveable and resizable lens is positioned over a page-based visualization of text. Areas outside the focal region are stretched to preserve visibility of the global context. Source: [50].

When targeting textual data for visualization, techniques based solely on fisheye views can meaningfully focus on at most a few documents at once. This makes them inefficient for dealing with the large-scale corpora described in Section 1.1. The concept of a “magic” lens was introduced by Bier et al. [6] to provide a powerful and straightforward focus+context approach. A moveable

see-through lens as the primary form of interaction enables a visualization’s user to focus on an area of interest without necessarily distorting its surroundings. Focusing on a region produces different effects depending on the implementation, including magnifying the area [2], adjusting its visual mappings or graphical properties [5] or dynamically filtering its contents [7], [24], [54]. Attesting to its versatility, the concept has even been extended to 3D spaces [56]. The full spectrum of techniques employing magic lenses is too vast to exhaustively list here, but all fundamental categories of data visualization (temporal, geo-spatial, graph, etc.) are represented [55].

However, rather few approaches exist where lenses are used to freely explore and navigate text collections. In 2011 Bosch et al. [9] demonstrated ScatterBlogs, a tool to explore geo-located micro blogging messages on a map of the world. This was later extended into a system for disaster management [10] based on real-time messages. As the lens is moved over the map, the contents of the user posts are extracted and visualized as a word cloud, with font sizes corresponding to a term’s prevalence.

2.3.4 DocuCompass

The DocuCompass technique [31] developed at the University of Stuttgart, Germany is the primary focus of the extensions proposed in later chapters. It is an approach to TextVis that combines the vector space models discussed in Section 2.2.3, the 2D document spatialization presented in Section 2.3.2 and a magic lens as the primary means of interaction with the visualization. DocuCompass itself is an adaptation of previous work done at the University like ScatterBlogs and a lens-based technique to explore the abstracts of scientific papers [32], as well as the CiteRivers tool for exploring citations [30]. It is intended primarily as a free exploration tool for large textual corpora.

In compliance with the standard InfoVis reference model of Section 2.1, DocuCompass first analyzes the input text library with NLP techniques. Tokenization splits each individual document at the granularity level of individual words, while Stanford’s CoreNLP library [43] is used for part-of-speech tagging, lemmatization and stopword removal. This rather advanced preprocessing compared to the simple stemming algorithms discussed in Section 2.2.2 is necessary not just to improve further analysis down the pipeline, but also because the tokens will be shown as part of the final visualization; having them transformed into a lexically incorrect word would only confuse and irritate the user. Each document is then associated with a very high-dimensional normalized *tf-idf* vector of its contents in preparation for 2D mapping.

DocuCompass employs the previously mentioned t-SNE [39] as its spatialization algorithm in order to reduce projection errors. It achieves this by first associating each pair x_i and x_j of vectors with a conditional probability function $p_{j|i}$ based on the likelihood that x_i would pick x_j as its neighbor in high-dimensional space by examining their Euclidean distance. Each pair of document points in 2D space y_i and y_j are similarly assigned a probability $q_{j|i}$. Afterward, a cost function that corresponds to the divergence between the two probabilities $p_{j|i}$ and $q_{j|i}$ over all document pairs (i, j) is minimized by gradient descent. The end result is that if two documents are similar in high-dimensional space, the probability of them being positioned close to one another in the final visualization is high.

The spatialization of documents as points in 2D space provides the “overview first” design principle recommended for visualizations [53]. The main method of interacting with the plot is through one or several circular magic lenses. When a lens is moved over a set of points, their fill color is changed and their border widened to indicate they are within its focus area. Furthermore, an ordered set of labels that is representative of their contents is placed around the

CHAPTER 2: BACKGROUND AND RELATED WORK

lens diameter. This is different than the ScatterBlogs approach, which places its word cloud visualization of document contents both in and around the lens, and is similar to the way Fekete and Plaisant’s excentric labeling [22] and its extension by Bertini et al. [4] work.

DocuCompass supports several different metrics that determine which terms should be drawn around the lens, including the previously discussed *tf* and *tf-idf*, but also the G^2 scheme. It compares the word frequencies of documents under the lens to those outside its focus area and selects terms that best differentiate the two. The advantages of this approach become most obvious when interacting with very uniform textual corpora where common terms would come to dominate the first few label positions if *tf* or *tf-idf* were used. Furthermore, research by Chuang et al. [17] has shown that the G^2 scheme creates summaries of a document’s contents that are more consistent with keywords manually selected by humans.

Beyond just moving the lens and showing labels, DocuCompass supports many other features tailored toward free exploration tasks. Those relevant to this thesis will be discussed in more detail in Section 3.1.

Chapter 3: Project Architecture

As outlined in Section 1.2 there are three goals to this master's thesis:

- creating an automated framework that determines the usefulness of the TextVis technique consisting of a scatterplot of document points (document points) and an interactive magic lens that summarizes their contents;
- adding static labels to the visualization that enhance the overview and guide the user toward further exploration;
- conducting a user study that evaluates the proposed extensions of functionality and analyzing the results.

This chapter discusses the overall architecture of the project and some of the design considerations that influenced the implementation.

3.1 Existing Implementation

The visualization technique that is the focus of this thesis is certainly derived from the DocuCompass application discussed in Section 2.3.4. However, evaluating the wider set of more advanced capabilities of DocuCompass accumulated over years of development is *not* the focus of this thesis. One of the goals in particular is to understand if the proposed extension of functionality (the static labels) is in itself a useful addition to the basic TextVis technique, i.e., the scatterplot of documents and the magic lens. Choosing to include the full scope of features of DocuCompass in the user study would only distract from that goal.

Because of this, for the purposes of this project an older codebase with reduced functionality was chosen to serve as the basis for the implementation. Since the overall capabilities of this system are noticeably different, it would be

both confusing and unfair to continue to refer to it as “DocuCompass”. Throughout this text, the project’s codename “LensMania” will be used instead to differentiate the two techniques.

Furthermore, because the goal is to extend an existing codebase and not to develop a brand new visualization, it is important to describe the functionality that was already present in LensMania. This will be the focus of the rest of this sub-section. Chapter 4 will discuss the original contributions of this project.

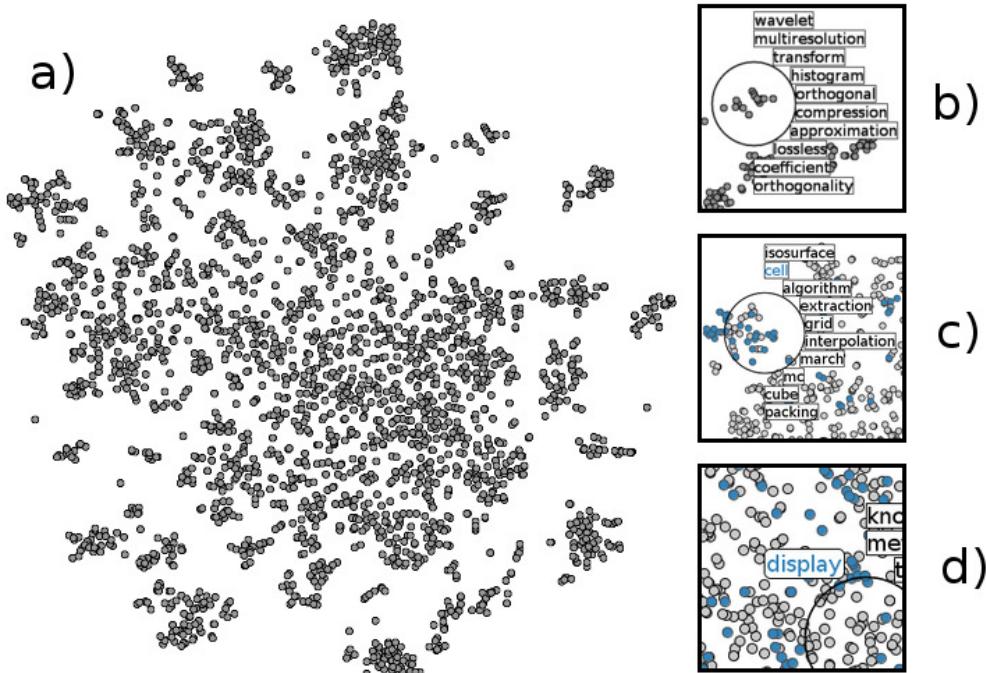


Figure 3.1: The LensMania TextVis technique. Shown are a) the 2D projection of documents, b) the terms shown when the lens covers documents, c) the term highlight functionality, d) the term pinning functionality. Source: own representation.

Much like DocuCompass, LensMania is a TextVis technique that is well-suited to free exploration tasks with no previous knowledge of the data set or a well-defined information need. It takes a text collection as input and projects the documents as points in 2D space (see Figure 3.1(a) for an example). The

main tool for interactive exploration is the magic lens, which the user may drag-and-drop to move and resize with the scroll wheel. When the lens is positioned over a set of documents, ten terms are displayed around its boundaries that summarize the documents' contents (Figure 3.1(b)). The term list is ordered by importance from top to bottom and there are no differences in font style, size or color.

One potential drawback of drawing the labels around the lens is that they occlude some of the documents next to it. However, the obvious solution of moving the summaries to a separate view would break the focus+context paradigm (see Section 2.3.3), thus forcing the user to constantly split their attention between two non-adjacent parts of the screen. Instead, labels continuously switch their positions to the side of the lens opposite the direction of its movement, ensuring that the most likely targets of further exploration are not occluded.

Hovering the mouse pointer over one of the labels selects it as a filtering term and changes its font color to reflect this. Any documents, whether within the lens' focus area or outside it, that contain that word become “highlighted” in the same color (see Figure 3.1(c)). This interaction is well-suited to an exploration scenario in which the user discovers a term of interest and wishes to investigate its presence or absence in other parts of the visualization. However, moving the lens to those regions requires drag-and-dropping the lens, which by necessity moves the cursor away from the highlighted term and resets the filtration. To reposition the lens and keep the document highlight active at the same time, the user may left-click on a label, which pins it to the side of the lens and keeps it permanently highlighted (see Figure 3.1(d)). At most one term may be pinned in this way, but the user may change the term at any time by left-clicking on a different label, or unpin the term by clicking on it.

3.2 System Overview and Technologies

This chapter describes the process by which LensMania transforms an input text corpus into a fully interactive visualization. Along the way, the technologies which are key to its functionality are discussed.

3.2.1 Prefuse Toolkit

The LensMania application was primarily developed using the Prefuse toolkit⁸ created by Heer et al. [29]. It is an open-source extensible software framework written in the Java programming language, released under the terms of the BSD license and targeted toward developing interactive visualization applications. Among its many features are a simple and well-documented application programming interface (API), an SQL-like expression language for querying and manipulating its data structures, a low memory footprint, full animation support and a library of functionalities often used in visualizations. It should be noted, however, that the toolkit does not target TextVis in particular and does not provide components for NLP.

Prefuse's design is directly modeled on the standard InfoVis reference model proposed by Card et al. [14], shown in Figure 2.1 and described in detail in Section 2.1. The relation between the model and Prefuse's internal packages and classes is depicted in Figure 3.2. Virtually every step of the visualization pipeline is well-supported: an input file with standardized data encoding is translated into an instance of the `Table`, `Graph` or `Tree` class depending on the type of data. A sequence of `Action` objects that map the table's values to visual variables can be chained and run to produce a `Visualization` whose constituent `VisualItem`'s can be further manipulated directly if necessary.

⁸ <http://prefuse.org/>

These are rendered onto a **Display** that supports view adjustments and interactions with the visual features.

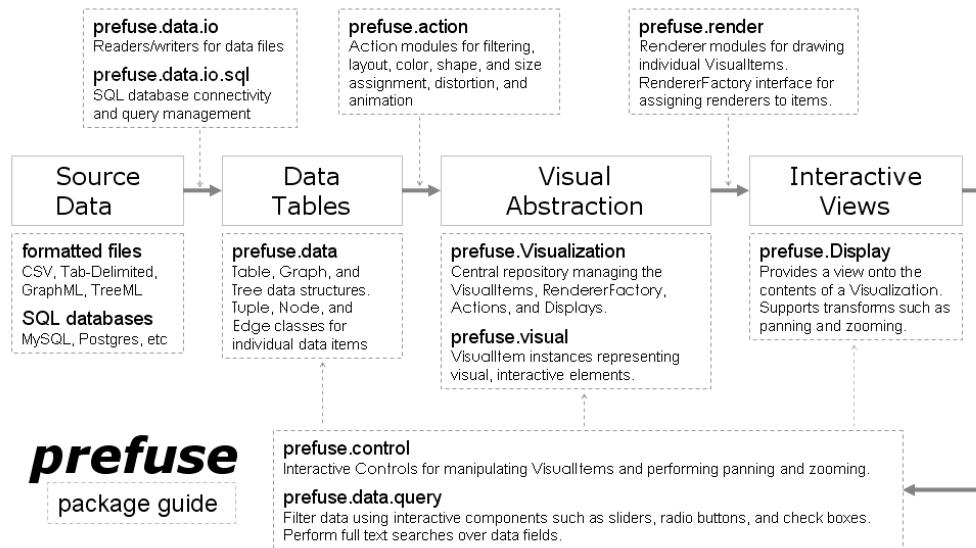


Figure 3.2: Overview of the Prefuse visualization toolkit. Depicted are the correspondences between its packages and classes and the InfoVis reference model. Source: <http://prefuse.org/doc/manual/introduction/structure/>

3.2.2 Application Architecture

LensMania's software architecture, depicted in Figure 3.3, is derived from the DocuCompass application. In order to support structurally different textual collections, the input is first transformed into a standard representation by an adapter component that also applies NLP techniques to the data to produce a *tf* bag-of-words. For the purposes of this thesis, the Visualization publications data set⁹ was used, consisting of more than 2,500 scientific papers produced between 1990 and 2014 whose abstracts form the textual corpus with which LensMania is evaluated. Part-of-speech tagging, lemmatization and stopword-

⁹ <http://www.vispubdata.org/site/vispubdata/>

removal (discussed in greater detail in Section 2.2.2) are handled by the Stanford CoreNLP library¹⁰. The resultant vector space model containing term frequencies is then weighted with *tf-idf* (see Section 2.2.3) and projected into 2D space using either t-SNE (Section 2.3.4) or PCA projection (both supported by the T-SNE-Java¹¹ library), producing the plot shown in Figure 3.1(a). In order to reduce application startup times, this representation is cached to a file on disk and re-used on subsequent runs. The bag-of-words vectors and the computed document positions are then stored inside a Prefuse Table instance, which is mapped to circular points on the display by means of runnable Action sequences.

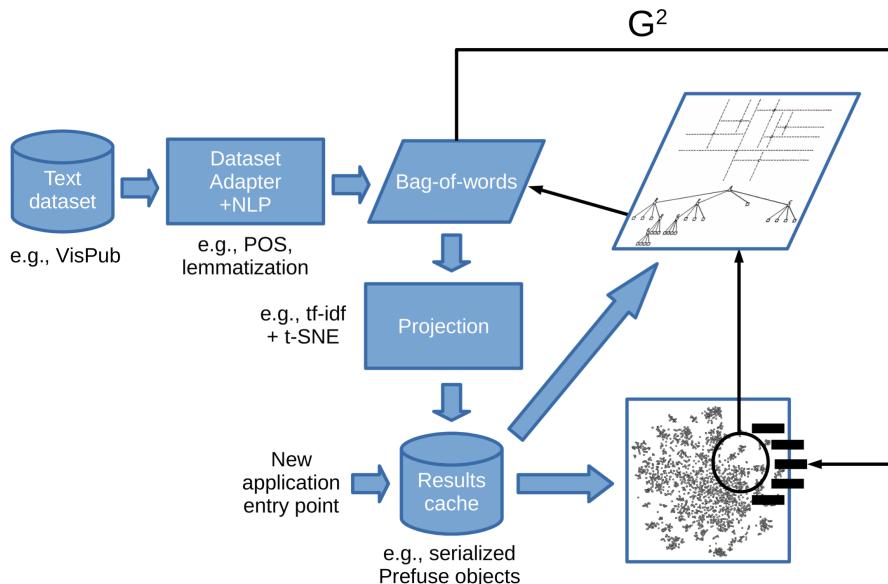


Figure 3.3: *LensMania’s application architecture*. A textual data set is processed by an adapter into a bag-of-words that are projected into 2D space, with results cache for faster application startup. Moving the lens causes a search in an efficient quadtree structure for documents underneath it. Weighted terms are extracted using G^2 metrics. Not shown: term highlighting requires a linear search in the bag-of-words. Source: own representation.

10 <https://stanfordnlp.github.io/CoreNLP/>

11 <https://github.com/lejon/T-SNE-Java>

When the lens is moved to a new area or resized, the application must fetch the documents that lie underneath it as quickly as possible so as not to break the user’s immersion with the technique. This is achieved by storing the document positions in a quadtree data structure [23] that allows for quick positional searches even for very large input data sized and keeps the overall visualization responsive. Once located, the bag-of-words vectors of the documents in the focus area are analyzed using the G^2 scheme (see Section 2.3.4) and the ten terms with the highest weighted frequencies are drawn around the lens. The label highlight functionality must linearly search the whole bag-of-words to find the subset of documents containing the term.

Finally, it should be noted that although the visualization publications data set, consisting of roughly 2,500 paper abstracts, is significantly smaller than the examples discussed in Section 1.1, it does exhibit some of the difficulties associated with large-scale data, like occlusion between document glyphs in the 2D layout, and satisfies the threshold of being prohibitively large to sieve through in a timely manner.

Chapter 4: Implementation

This chapter describes the design considerations and choices made while implementing the two principle contributions of this thesis – an automated framework for the quantitative evaluation of the LensMania technique and static labeling of the 2D layout of documents.

4.1 Automated Framework

The need for a system that works in the background to gather statistical data about the basic TextVis technique consisting of a scatterplot of docupoints and a magic lens that extracts and shows keywords arises from some of the choices made during the conceptualization of LensMania. In particular, even if one assumes that the high-dimensional vector space model produces a good summary of the contents of a textual corpus, projecting the documents into 2D space introduces both new groupings of data points and errors inherent in any type of dimension reduction. Other design decisions like the diameter of the lens and the number of terms it extracts may also influence the technique’s utility. A quantitative evaluation of these two properties in particular may improve confidence in LensMania’s usefulness for exploration tasks.

4.1.1 Automated Logging

An automated framework for gathering the necessary data was developed that runs separately from the normal flow of user interaction with the technique. The display’s size was fixed to 800×800 pixels. A magic lens is placed at the top left corner of the application window and programmed to visit every single point of the visualization in succession (see Figure 4.1). At each position information about which documents it covers and which labels it shows is

logged into a human- and machine-readable database. The entire process is repeated for 40 different lens diameters so as to incorporate the resize-by-scrolling functionality into the logged information. Although the application normally shows only the first ten most prominent terms, the framework records the first twenty to evaluate the effects of increasing this default value. The simulated lens movement which exhausts all possible screen positions and a representative sub-sample of lens sizes is not intended to directly model a human's interaction with LensMania. Instead, it is a metrics-gathering framework for quantitative analysis of the technique. Figure 4.2 shows the extended software architecture of the application. Note how the logging is kept separated from the other components, indicating it is not performed as part of a regular user session but as a separately run process, much like the preprocessing and caching of input data.

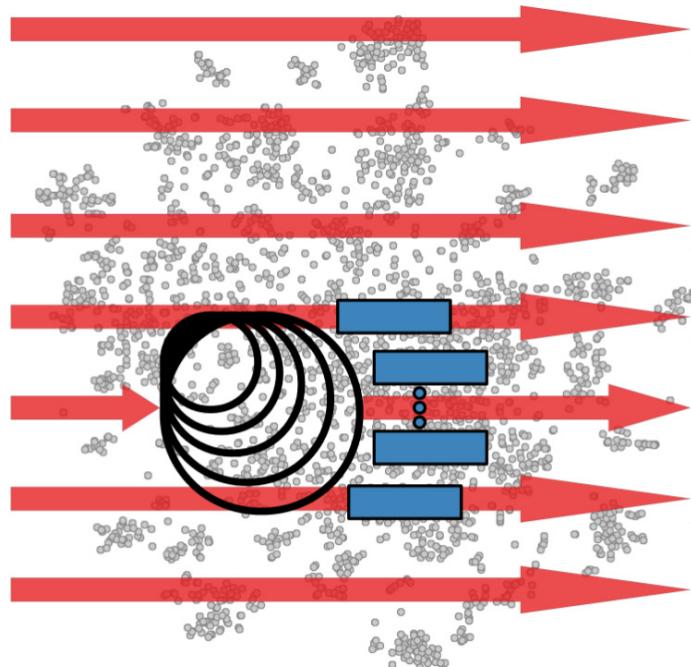


Figure 4.1: Automated Logging Framework. Lenses with varying sizes are continuously moved through every point in the visualization. At any given position the covered documents and twenty terms are recorded into a database for further analysis. Source: own representation.

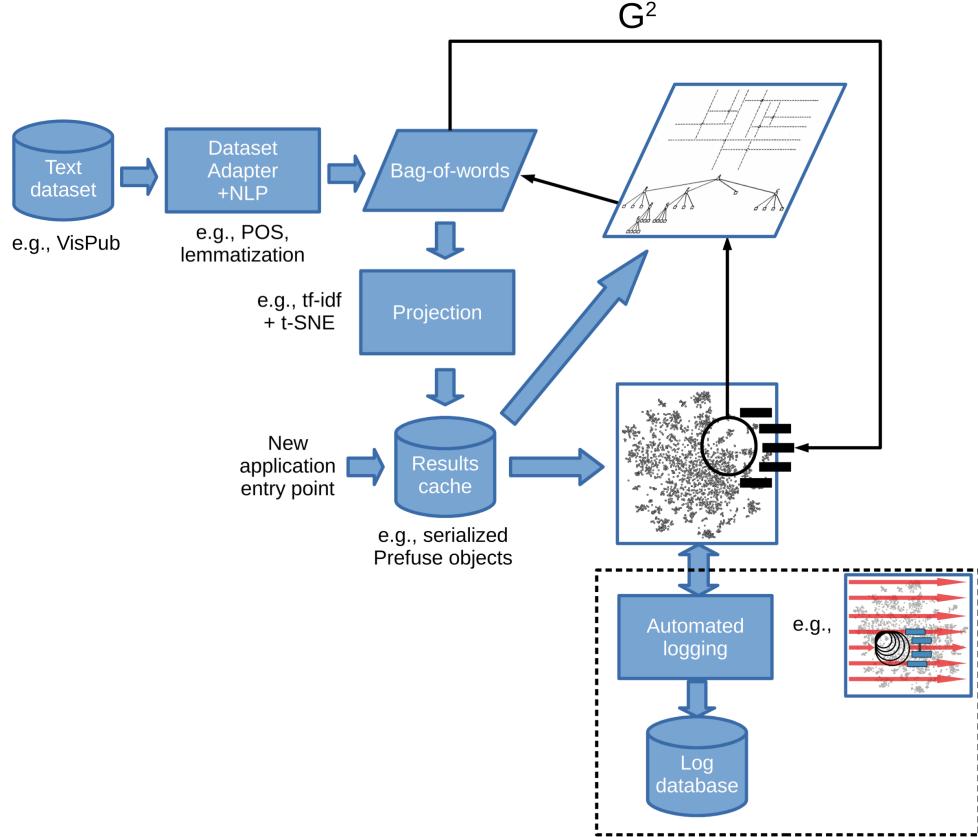


Figure 4.2: LensMania’s application architecture extended with logging. The newly added logging component is not part of a regular user session. Source: own representation.

4.1.2 Size of Logging Database

In total, $800 \times 800 \times 40 = 25,600,000$ samples and 28 gigabytes of uncompressed data were recorded by the automatic logging framework over the course of several days. This in itself presents a large-scale data analysis challenge. Even filtering out all lens positions in which no documents were covered and thus no terms were displayed does not noticeably shrink the data set. Reducing the number of samples by skipping every few display positions does ac-

celerate both the gathering and the analysis of data, but no longer exhaustively covers all points a user may wish to drag the lens towards.

The straightforward and often used approach of storing the information in an SQL database and manipulating it with standardized and highly optimized queries was attempted but found to be prohibitively slow. In the end, a set of self-developed and parallelized programs written in the C programming language were created, specifically tailored toward the analyses presented in the following sections.

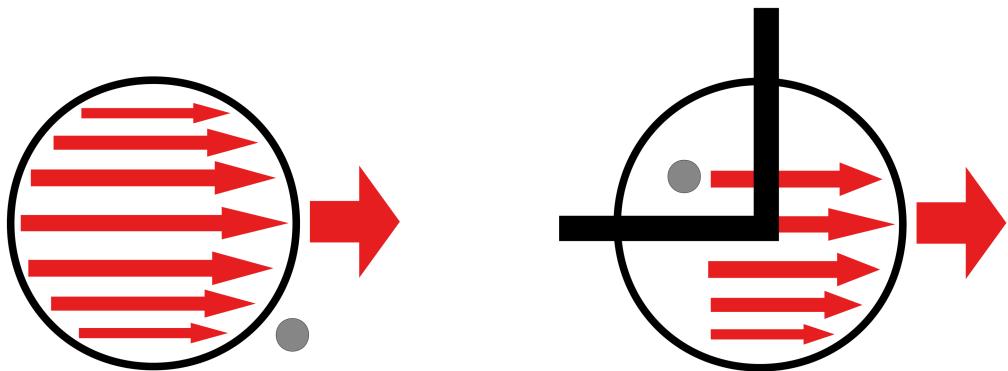


Figure 4.3: Document coverage model. In a “perfect” scenario (left) the lens covers the documents exactly πr^2 times. However, if the document is near a border or corner, the lens will cover it fewer times. The restriction that the lens center cannot leave the display boundaries is either enforced by the technique or occurs naturally in actual use. Source: own representation.

4.1.3 Document Coverage

One question of interest that arises is the probability that an individual document falls underneath moving lenses of various sizes, i.e., how often it is considered for term extraction using G^2 metrics. Consider the scenario in Figure 4.3(left). A lens that visits every display position will cover the docupoint with each of its constituent pixels exactly once. Consequently, the expected cover-

age for every single document is the area of the lens πr^2 , while the probability of coverage is that divided over all 800×800 possible positions. However, a complication arises that prevents this “perfect” scenario from occurring in practice. Near the borders and corners of the visualization not every pixel of the lens’ area covers all documents equally because it is prevented from being completely dragged out of the display (see Figure 4.3(right)).

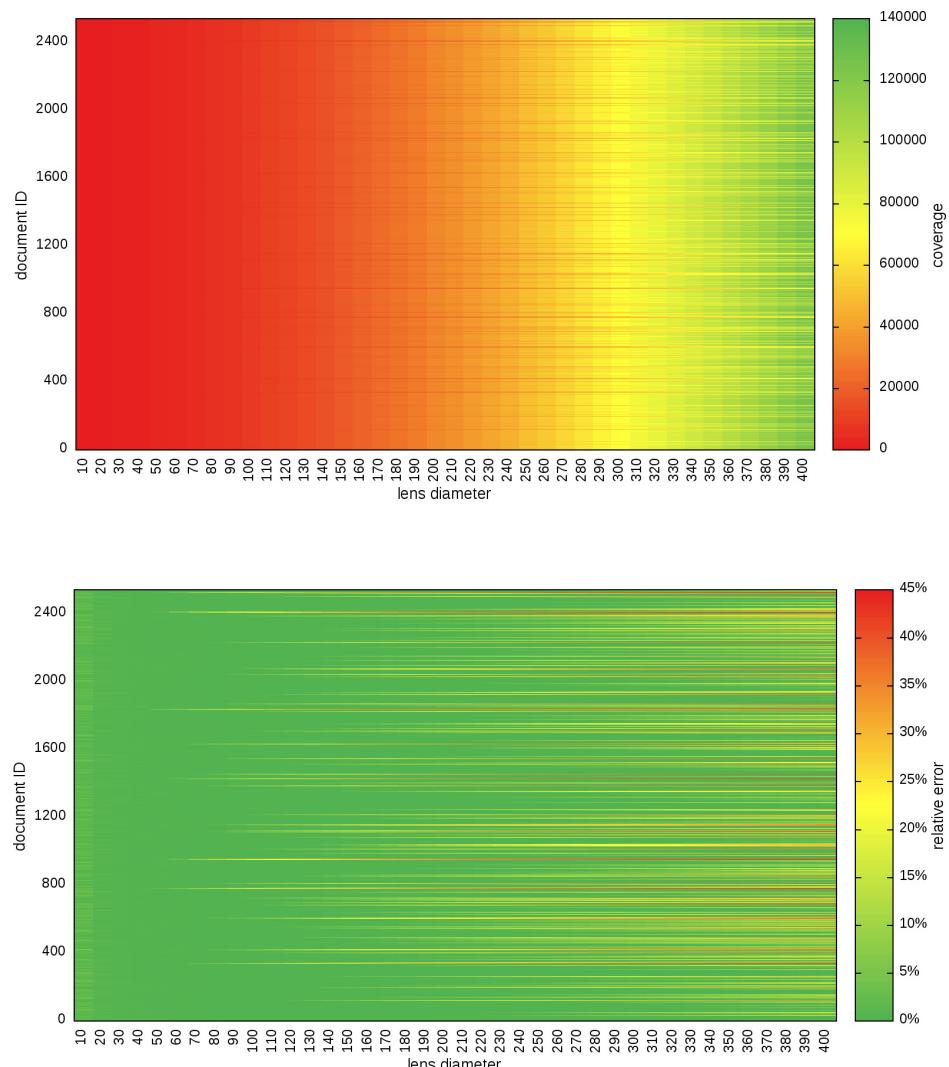


Figure 4.4: Actual document coverage (top) and relative error (bottom). Deviations from the expected value grow higher as the lens diameter is increased. Source: own representation.

The data gathered by the automatic logging framework can be used to pinpoint such irregularities. Sieving through the database and extracting the exact number of times a document was covered produces the results shown in Figure 4.4(top). As predicted, increasing the lens radius also improves the coverage, but not equally for all documents. Figure 4.4(bottom) depicts the relative error from the expected value of πr^2 , which clearly increases with the lens diameter.

4.1.4 Document Prominence

While all documents from the input data set are similarly depicted as a single point in the final visualization, it is not at all obvious if they are equally well-represented in the term list shown to the user. This section discusses several metrics for modeling this document “prominence” within the LensMania visualization depending on what lens size is used and how many terms are shown (referred to as the “variables” in this section).

Choosing to employ more than ten labels at once can obviously only increase the prominence. To account for this bias, all of the models in this section are scaled down by dividing them with the number of terms. This makes sense in the context of the LensMania application – showing too many labels around the lens leads to greater occlusion of the docupoints and may serve only to distract the user from the brief summaries the technique relies on.

Altering the lens’ diameter, on the other hand, produces two conflicting effects. As demonstrated in the previous section, using a larger lens increases the probability that a document is considered for term extraction. At the same time, however, it must compete with more documents for a chance to be represented in the term list. Because of these opposing influences, the choice was made not to scale the models based on lens diameter.

One simple way to calculate prominence is to consider all instances in which the lens covered a particular document and count the number of terms extracted from those positions that can also be found in that document’s personal bag-of-words. This sum can be plotted against the two primary variables whose influence on LensMania is of interest – the number of terms to display and the lens diameter. In fact, up to four different views can be presented of such a model:

- a document view, in which the prominence values for a single document are shown plotted against the lens sizes and term limits (Figure 4.5);
- an average view, in which the arithmetic means over all 2,500 documents in the collection are plotted as above (Figure 4.6);
- a “best” view, in which the 2D layout of documents is colored based on the variable combinations that produce the *highest* prominence (or, a “worst” view if the reverse information is desired) (Figure 4.7);
- a “fixed” view, in which the lens size and number of terms remain fixed and the docupoints are colored comparatively to one another based on their prominence (Figure 4.8).

The goal of these visualizations is to support decision making in regards to selecting meaningful defaults for the two variables. The document view and average view plots are interpreted in exactly the same way, with combinations of lens size and term limit that provide higher prominence values colored green. The two scales differ significantly in their magnitude, suggesting this particular document is likely to be a non-representative outlier case.

Much like average view, best view can be used to select a combination that guarantees higher prominence over the largest number of documents. Fixed

view, on the other hand, identifies documents that are particularly poorly represented in the extracted terms for a chosen pair of variables. This knowledge can be exploited by visualization applications to offer additional guidance during interaction with those areas of the plot.

A drawback of this basic model is that it counts every term with correspondence in the vector space model equally, regardless of how descriptive that term actually is for a particular document. A natural extension of this approach is to instead sum up the corresponding tf-idf frequencies for a document-term pair.

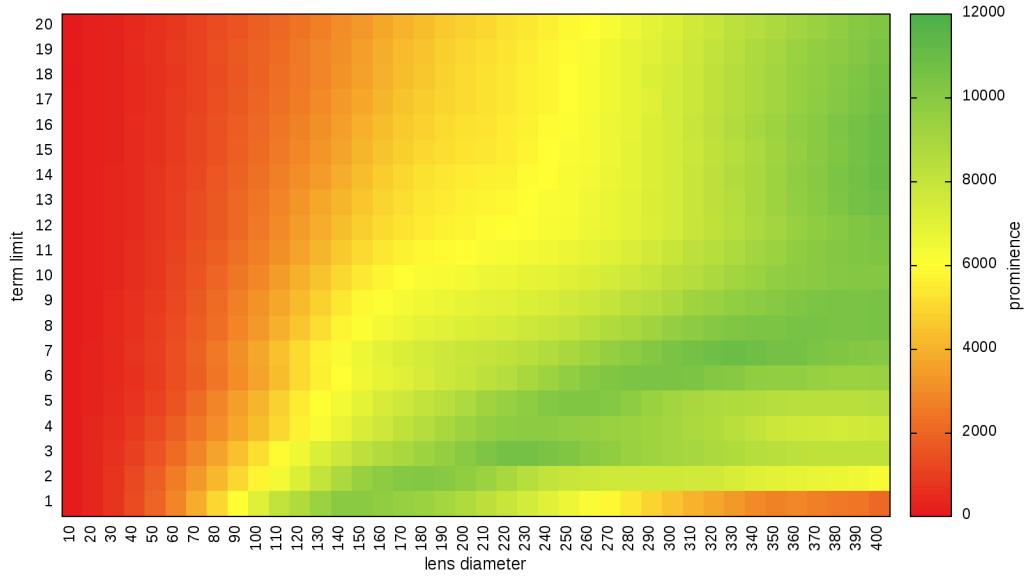


Figure 4.5: Basic prominence model: document view. Particularly good variable configurations for an individual document are shown in green. Source: own representation.

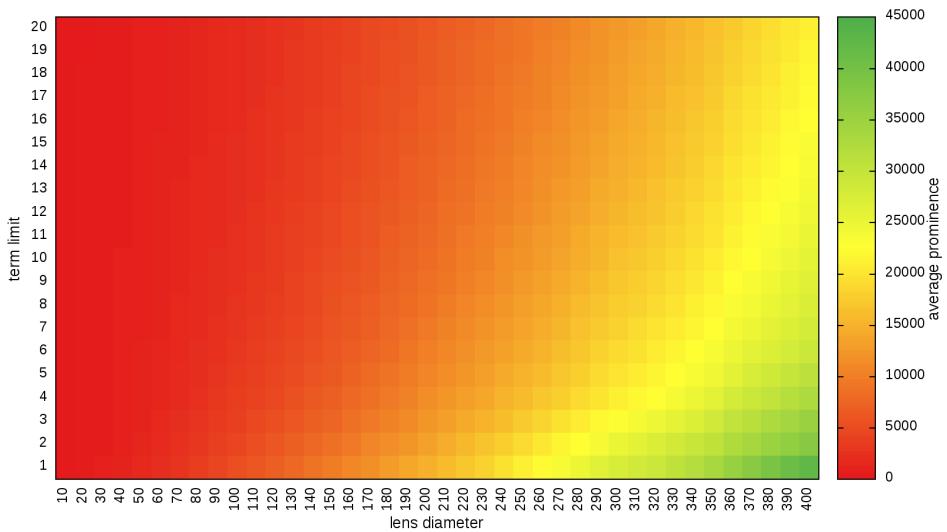


Figure 4.6: Basic prominence model: average view. Same interpretation as the document view, but with values averaged over all 2,500 visualization documents. Note the different magnitudes of the scale. Source: own representation.

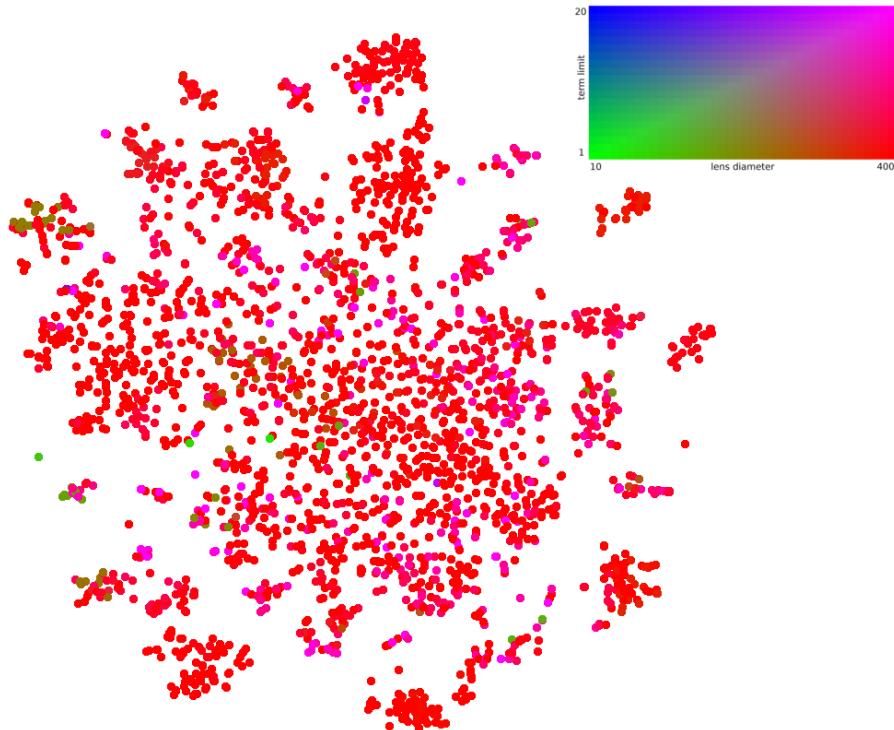


Figure 4.7: Basic prominence model: best view. As suggested by the average view, under this model most documents prefer higher lens sizes and lower term limits. Source: own representation.

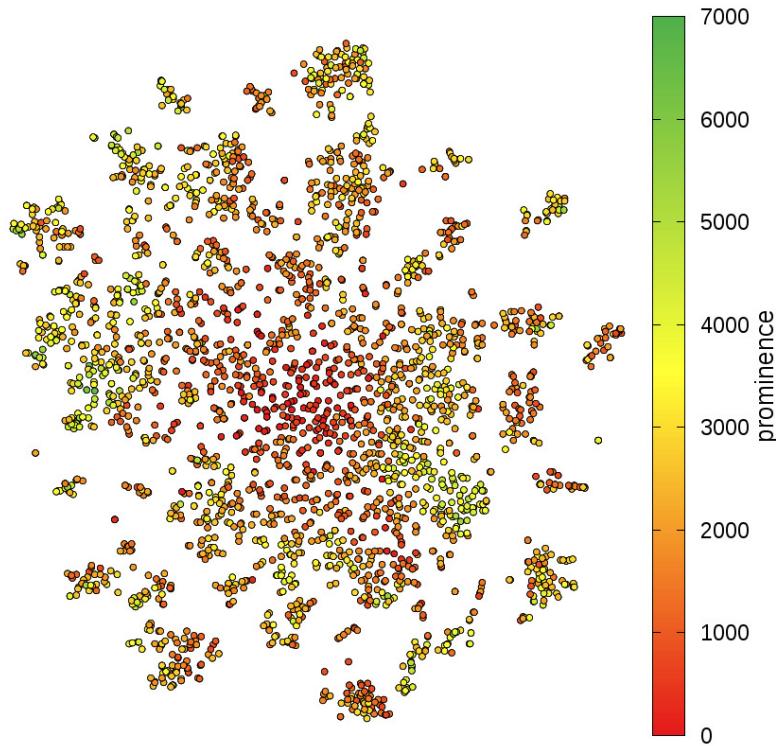


Figure 4.8: Basic prominence model: fixed view. Choosing to set the lens size at 100 and the term limit to 10, one can examine what effect those values have on areas of the plot. Notice that that magnitude of the scale is lower than the one in average view because of the non-optimal choice of values. Source: own representation.

4.2 Static Labeling

The paper by Heimerl et al. [31] that introduced the DocuCompass technique speculates that adding static labels to the 2D layout of documents may increase its usefulness as a free exploration tool. Truthfully, a user presented with an unfamiliar textual corpus will as a first step usually spend some time performing a linear search with the magic lens through its contents until an area of interest is found. If labels which characterize the documents were placed in advance within the visualization, these may guide the user toward such regions more quickly. This section of the thesis discusses LensMania's ap-

proach to static labeling and showcases several prototype implementations. The most promising of these is evaluated by means of a user study in Chapter 5.

4.2.1 Document Clustering

While in theory any part of the Visualization publications data set, including metadata, can be used to generate the necessary labels in any number of ways, in practice users would probably expect that the same set of keywords are associated with an area of the display that would be if the magic lens was positioned over it. It makes sense, therefore, to base static labeling decisions on the 2D layout of documents produced with t-SNE and on the term extraction done by G^2 . Clustering can be used to select document subsets within the visualization which are important enough to be statically labeled.

LensMania already relies on the distance between docupoints as a metaphor for the similarity between texts. It makes sense to also base the clustering decision on these distances. Hierarchical clustering is a well-known, if somewhat expensive approach. The “bottom up” variation of the algorithm looks like this:

- (1) *Start with each point in 2D space in its own separate cluster;*
- (2) *Find the two clusters physically closest to one another and merge them;*
- (3) *Repeat (2) until all points belong to a single cluster.*

This can be conceptualized as a tree-like structure with individual data points as its leaves and increasingly larger clusters as its nodes, with the root node corresponding to a supercluster of all points. A choice can be made to cut the tree at an arbitrary distance from its leaves, i.e., to halt the algorithm after the desired precision is achieved. Because clusters consist of many points, it is necessary to redefine the distance measure between them. Three popular approaches are:

- using the minimum distance between any two points not in the same cluster (called single-linkage clustering): $\min\{\text{dist}(a,b), a \in A, b \in B\}$
- using the maximum distance instead (complete linkage clustering):
 $\max\{\text{dist}(a,b), a \in A, b \in B\}$
- using the distance between the centroids of the clusters.

LensMania employs a slight variation of single-linkage hierarchical clustering that uses a threshold value to control the clustering precision. The algorithm works as follows:

- (1) Define m , the threshold value above which a distance is considered too great for clustering two points together;
- (2) Start with each point in 2D space in its own separate cluster;
- (3) For each point i , find those points whose distance to i is less than m and merge them and their clusters with i and its cluster.

In other words, if two points can reach one another by means of “hops” (between other points only) individually no greater in distance than the threshold value, then they are clustered together. The algorithm intentionally places no restrictions to how many points a cluster may contain and no point is left unclustered (although single-point clusters can exist). The value of m can be decreased to allow only very localized clusters or increased to bridge wider gaps in the projection.

During the development of the static labeling enhancement an alternative density-based clustering approach was also explored. Empty regions of the visualization are assigned a value of zero, while all docupoints are considered to have a value of one. A normalized 2D Gaussian kernel of the form

$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ is convolved with the document layout depicted in

Figure 3.2(a), producing a smooth density map (Figure 4.9) that can also be thought of as a heightmap. A threshold value can then be selected that serves as a 3D cutoff plane: documents under it were spaced too far apart and are not clustered at all; documents over it were crowded together to form mountaintops that can be directly interpreted as clusters. This can be extended to a scheme that takes not just the 2D positions of documents into account when clustering, but also their textual contents, in the way the SPIRE system’s Galaxies visualization is created (see Section 2.3.2 and [59]).

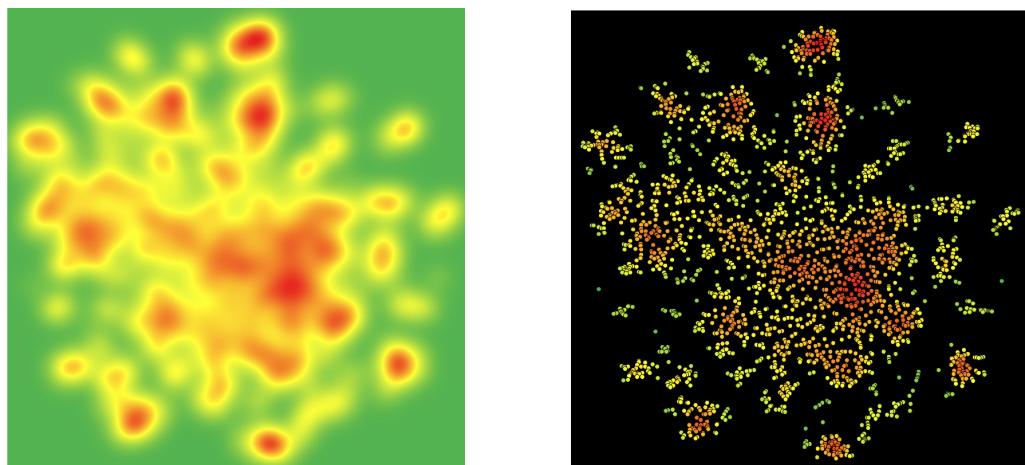


Figure 4.9: Density-based clustering. (Left) A 2D Gaussian kernel is convolved with LensMania’s layout, producing a map of regions with high concentrations of documents. (Right) The values on the left mapped to the docupoints. The black background is for contrast only. Source: own representation.

A discussion of the many other clustering techniques in existence is beyond the scope of this thesis.

4.2.2 Non-Occluding Static Labeling

Once clusters of tightly packed documents have been identified, the G^2 scheme of Section 2.3.4 can be applied to their contents to extract an arbitrarily long list of terms ordered by importance. Positioning the static labels within the visualization is a challenge in itself; ideally, the words should not overlap with each other and disrupt as few other features as possible.

A first attempt at including static labeling within LensMania was based around positioning the terms so that they do not occlude the document points at all. This may be understood as a point-feature label placement problem common in cartography and known to be NP-hard [16]. Some approaches to solving it include:

- exhaustive search algorithms that place labels in the first available position, but backtrack to choose another if that placement decision makes the proper placement of later labels impossible;
- greedy algorithms that, again, take the most opportune position first but without regard for future label placement
- linear programming that maximize an objective function corresponding to preferences in label placement.

LensMania employs the particle-based labeling algorithm by Luboschik et al. [38] which is an example of the second category of approaches above. Because finding *any* viable position for a label, let alone an optimal one, is not guaranteed by the method, the order in which terms are placed is significant.

The first prototype of static labeling developed used the following positioning algorithm:

- 1 *Identify the set of n biggest clusters and computer the average x and y coordinates over all their document points;*

- 2 Extract the first m terms from each cluster in the set;
- 3 For each cluster in the set, ordered from largest to smallest, do:
 - 3.1 For each term, from the most to the least important one, do:
 - 3.1.1 Position the label at the average coordinates of its cluster;
 - 3.1.2 Move the label along a spiral pattern until either a viable non-occluding position is found or the label leaves the screen boundaries; if the latter is true, remove the label altogether;

The algorithm therefore only applies labels to a select few clusters and then only places several labels. To understand why this is so, let us consider the visualization of its results in Figure 4.10. While it is true that the labels are positioned physically close to the clusters that created them, it may not always be obvious which term comes from which cluster. To create an association between the two, both the document points and the label backgrounds are set to the same color (points outside the largest clusters are colored gray). While this mapping is intuitive and easy to grasp, there are not enough contrasting colors to support more than 10–15 clusters at once. Also, increasing the number of labels displayed per cluster would quickly fill out the entire screen and force the discarding of important words from smaller clusters.

Step 3.1.2 of the algorithm deserves special attention. Because non-occlusion is a stated goal of this prototype, viable positions are defined as those in which the label to be placed does not overlap with any document points or other labels. Checking if this is true for the low number of already existing labels is a non-issue, while the quadtree data structure discussed in Section 3.2.2 can be quickly queried for potential overlap with documents. The spiral pattern formula presented in the original paper allows for various granularities depending on precision and speed requirements, yet the algorithm remains greedy in

nature: the first viable position is always assumed with no regard for possible non-optimal placement in the future.

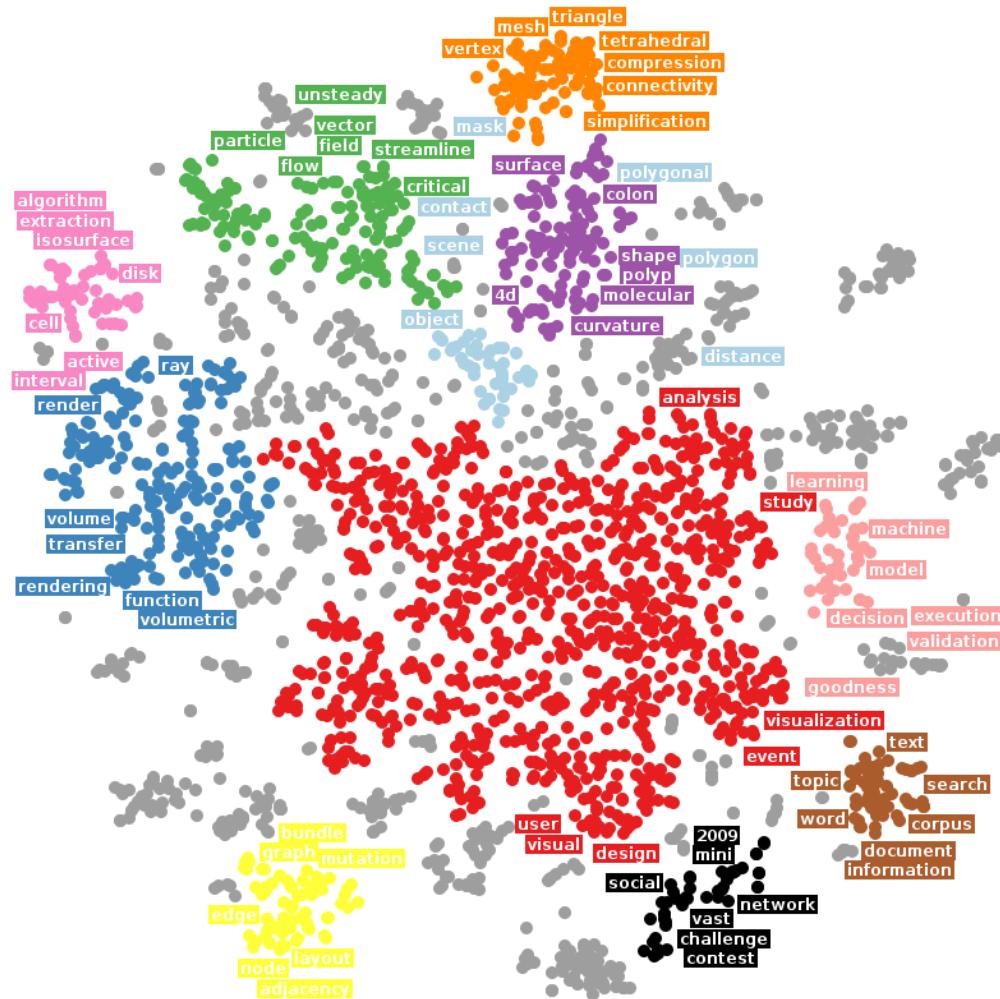


Figure 4.10: First static labeling prototype. Color is used as the primary means of associating a docupoint to the labels. Source: own representation.

Because of the lack of overlap with document points, the labels do not get in the way of the magic lens' purpose as the primary method of exploration. Therefore, no interaction between lens and labels was deemed necessary in this first prototype. An early improvement to the algorithm outlined above was to

check if a label wanders too far away from the boundaries of the cluster it was supposed to represent, in which case it is considered a hindrance to the overall visualization and is discarded. Also, to address the lack of clear indication as to which of a cluster’s labels are the most important ones, their order of extraction was mapped to a decreasing font size, similar to how a small word cloud visualization (Section 2.3.1) might look.

The most readily identifiable drawback of the proposed technique is the small number of clusters it can support due to the mapping to color. This influenced the threshold value provided to the clustering algorithm discussed in the previous section – so as to provide coverage of a meaningful subset of documents, the few clusters actually selected for labeling had to be kept very large. This often meant the G^2 scheme would extract overly broad and non-helpful words, while users would be surprised that their average-sized magic lenses displayed none of the labels extracted from the bulky clusters. Simply put, the technique was non-versatile and would not scale well.

A second prototype was developed that tried to address the primary issue. It removed the color mapping altogether and decreased the cluster sizes, while significantly increasing the number of clusters chosen for static labeling (see Figure 4.11). The varying font sizes feature was removed and labels were made smaller in general to facilitate drawing more of them on the screen. The association between labels and clusters was now achieved by coloring both in a highly contrasting color when the user hovered over a term. As before, no interaction between magic lens and labels was deemed necessary.

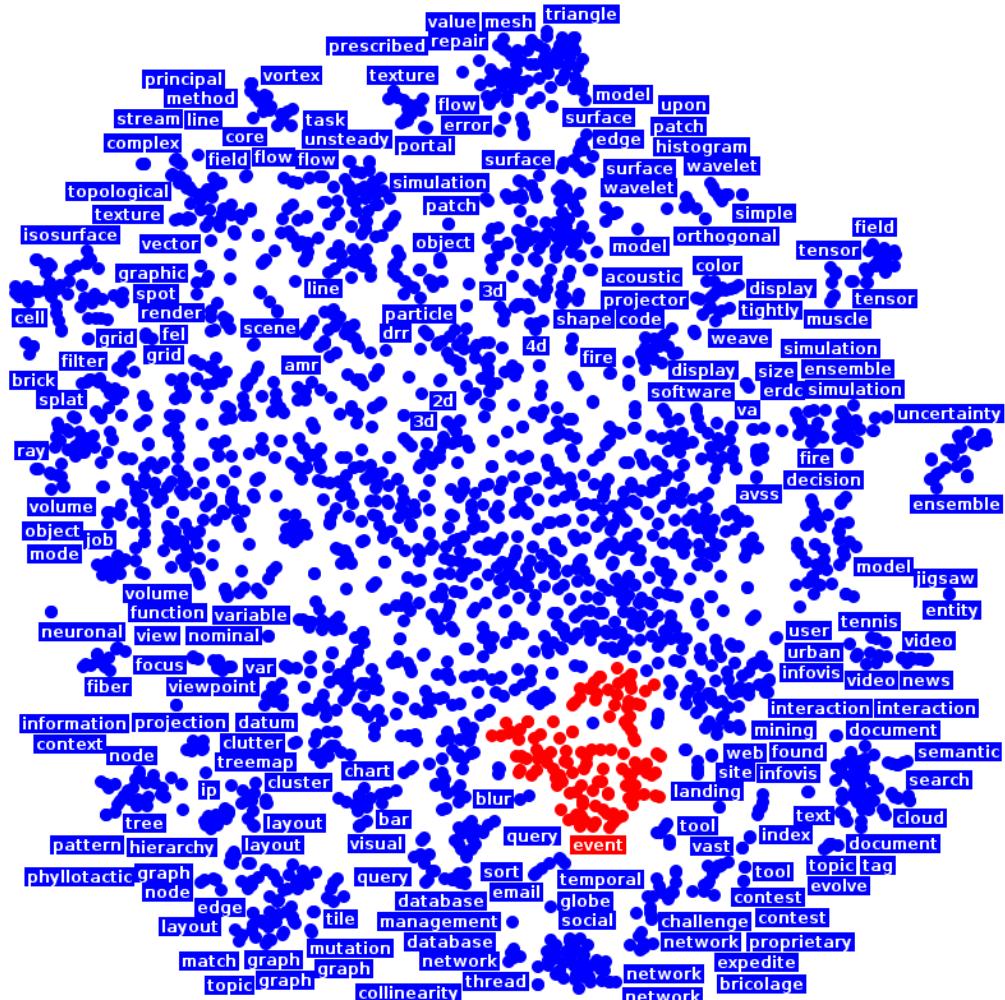


Figure 4.11: Second static labeling prototype. Hovering over a term reveals the cluster of documents it represents. Source: own representation.

While being somewhat of an improvement over the first prototype, this version brought its own set of problems. Having such a huge number of documents and labels on the screen all colored similarly would certainly appear intimidating to users, while the smaller font sizes were detrimental to legibility. The smaller clusters provided better correspondence between static labels and lens terms, but the words on screen tended to displace one another from more optimal positioning.

4.2.3 Static Labeling with Occlusion

Because of the drawbacks to the approaches discussed in the previous section, a decision was made to develop a third prototype, this time with static labels made slightly transparent and positioned directly above the clusters they summarize (see Figure 4.12(left)). While this does occlude some of the data patterns in the 2D document layout, it provides a much more straightforward and logical way for users to map a label to the area of the plot from which it comes. Varying label fonts sizes to indicate importance was re-introduced as a feature.

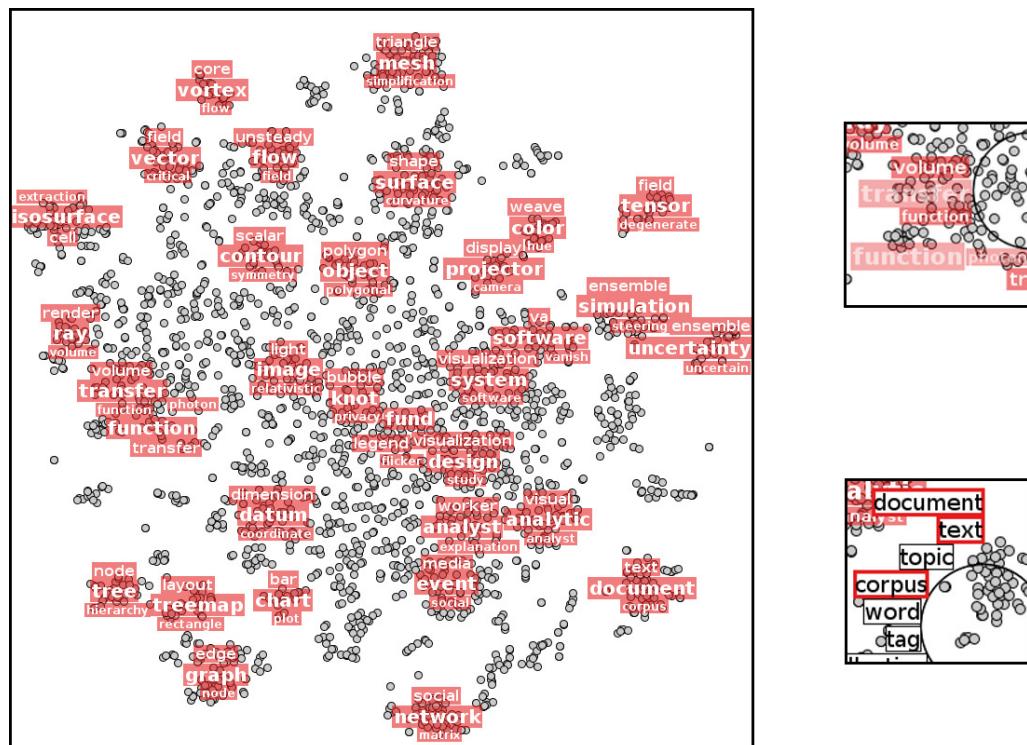


Figure 4.12: The final static labeling prototype. (Left) Positioning of the static labels over the document points. (Top right) Displaced static labels have their transparency increased. (Bottom right) A displaced label that also occurs in the term list is hidden and the term is outlined. Source: own representation.

The changed approach also made necessary the addition of meaningful interaction between the magic lens and the labels. As the lens moves to interact with the documents, static labels that would otherwise obstruct its view are pushed away, revealing the patterns underneath once more. But should the lens move away, the labels return to their default positions, continuing to provide guidance to the user about areas of the visualization not currently in focus. However, overlap between labels should still not be allowed to occur to keep them legible. To this end, if a label that is pushed out by the lens comes to occlude other, it pushes the later away as well in what can be likened to a transfer of momentum. To get out of sight quickly during exploration with the lens and to indicate to the user that they are no longer representative of the documents directly underneath them, displaced labels have their transparency increased (Figure 4.12(top right)). Finally, if a pushed-out label should become duplicated in the term list around the lens, it disappears from the main visualization view to reduce clutter and highlights the term in a similar color to its own (Figure 4.12(bottom right)) (this was dubbed the static label hiding feature). Its original position is restored once there is no longer any duplicate information. The initial label positions are still chosen by the spiral-based algorithm to ensure that the most important ones are prioritized for placement and that there is no overlap.

The method by which labels are pushed out by the lens and later return to their default positions without occluding each other deserves some attention here. It is dubbed the pull-push algorithm. Whenever the lens is moved or resized, i.e., when the possibility of interaction with static labels arises, *all* labels are first reset to their initial locations and states. This simulates a constant pull towards their respective points of origin. At this point, a check is made to see if any labels intersect or lie within the new focus area. If so, they are compared to

the list of terms around the lens (if any) and if a duplicate is discovered the label is made invisible.

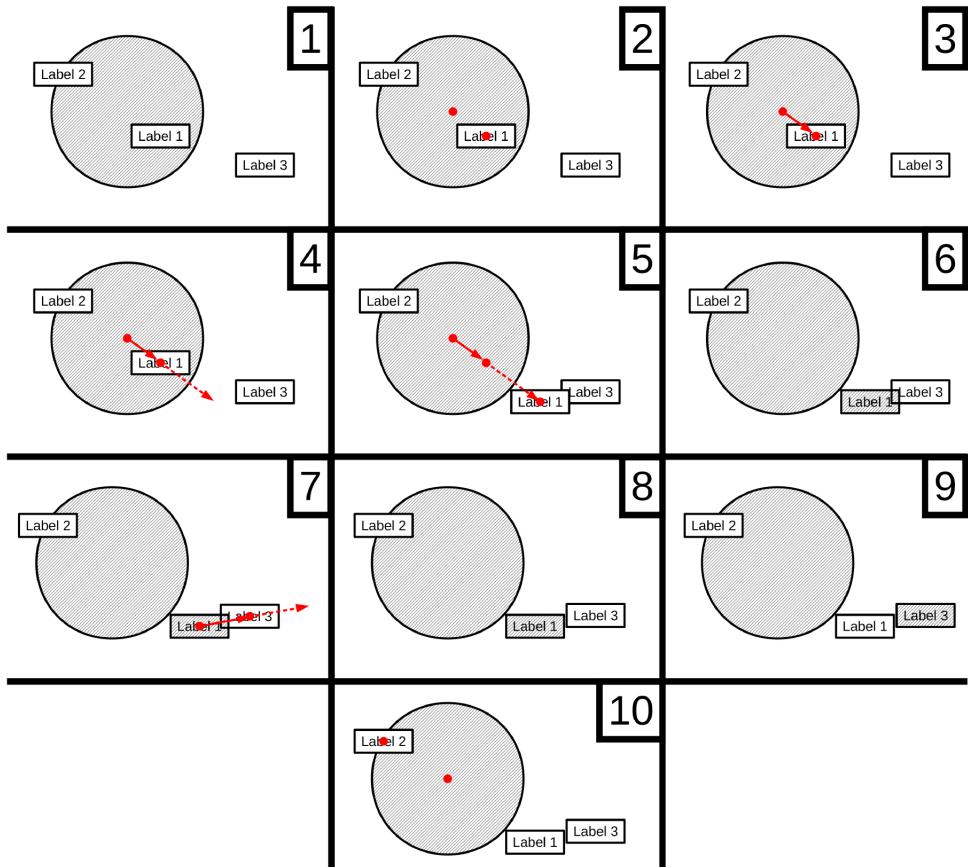


Figure 4.13: Visualization of the push-pull algorithm. Source: own representation.

The push part of the algorithm is depicted in Figure 4.13. The vector that connects the centers of the magic lens and a label is computed and normalized in its size to allow for smoother movement, then continuously applied as a displacement to the label's position until there is no longer any overlap. A label that has been moved in this way is marked as “dirty”, i.e., a re-evaluation of its area is needed to see if intersections with other labels exist. If that is the case, those are displaced in the same manner, also marked as “dirty” and so on until

no more overlaps exist. Finally, if the algorithm detects that a label has been pushed to one of the borders, the displacement vector is made parallel to it to slide the label along its length instead of pushing it out.

Out of the three static labeling prototypes discussed in this chapter the last one showed the most promise and had the fewest obvious flaws. For these reasons it was selected as the basis for an evaluation by means of a user study.

Chapter 5: User Study

A scientific evaluation is a critical part of any new InfoVis technique. It helps promote good ideas and increases confidence in the effectiveness of existing ones [25]. The final goal of this master’s thesis – the conduction of a think-aloud user study to evaluate the LensMania application and its approach to static labeling – is the subject of this chapter, as well as an analysis of the results and a discussion of lessons learned.

5.1 Research Hypotheses

The main goal of the user study was to evaluate whether or not the Lens-Mania application is useful for its intended purpose – free exploration tasks of large text corpora. Of additional interest was if and how static labels contribute to this. To separate the two questions, a between-group user study was envisioned, with participating users assigned at random a tool to interact with: either the initial codebase of Section 3.1 (hereafter referred to as the “basic technique”) or the LensMania variant with occlusion-based static labeling from Section 4.2.3 (the “advanced technique”). The following research hypotheses were formulated:

1. That users would perceive the advanced technique as easy to learn, subjectively satisfying and useful for exploring large textual data sets.
2. That the higher occlusion in the advanced technique would not be a major hindrance to exploration.
3. That users would employ different approaches to solving the tasks when using the basic technique as opposed to the advanced technique.

4. That the advanced technique would help users answer general and specific questions about a data set more quickly, accurately and with greater confidence than with the basic technique.

5.2 Structure of the Study

To test the research hypotheses, two questionnaires were created, one each for the basic and advanced techniques (see Appendices B and C). They consisted of:

- a page asking general statistical questions like age, gender and background with visualization techniques;
- a free exploration task in which the users are asked to think aloud while interacting with either technique;
- a search task in which the users search for a clustering of documents about social networking;
- a set of follow-up questions that asks for the user's opinion of each of the interaction techniques made available to them, as well as more general questions about the usefulness of the system.

Also, a consent form was created that instructed the participants on the nature of the data to be recorded and explained them their rights, in particular the right to quit at any time (see Appendix C).

5.2.1 User Logging

Much like the automated framework discussed in Section 4.1, an automatic logging of users' interaction with the technique can provide insights into its usefulness as a tool for free exploration. It was therefore incorporated as the final component (see) of the software architecture first described in Section

3.2.2. All of the interactions supported by LensMania are included in the logging; in particular it timestamps and records:

- all positions to which the lens is dragged;
- all resizing of its radius;
- the documents it covers and the term list it shows;
- when a term is highlighted or pinned and which documents become emphasized as a result;
- which static labels are displaced by the lens;
- when a static label is hidden due to duplication in the term list.

Because the user study features a think-aloud session during interaction with the technique, a voice recording by microphone is also made, as well as a screen capture, in order to contextualize what the user is talking about with what is happening on-screen¹².

5.2.2 Protocol for Conducting the User Study

The user is welcomed to an evaluation room, seated in front of a computer monitor and a microphone and asked to switch off their mobile phone. The researcher thanks the user for their willingness to participate and then either reads aloud or allows the user to read the consent form for the study. Afterward, the user is allowed to ask explanatory questions and must explicitly reaffirm their willingness to proceed with the study with their signature. They are then handed the page with general statistical questions to fill out.

¹² Only the logging of user interactions was implemented in the LensMania codebase itself. Voice and screen recording was done with external software.

The user is then randomly shown either the basic or the advanced technique with a test data set. The researcher explains the complete functionality of either system (basic or advanced), including:

- explaining that the points represent textual documents, with physical closeness roughly corresponding to greater semantic similarity between them;
- demonstrating to the user how to add a lens to the visualization and how to move and resize it;
- showing the term list around the lens to the user and explaining how it is representative of the documents currently under the lens;
- demonstrating the term-highlight and term-pinning functionality;
- if demonstrating the advanced technique: pointing out the static labeling hiding and push-out features to the user.

The participant is then asked explicitly for any questions they have about the technique or if they require repeated demonstrations of functionality and is allowed brief interaction with the technique to learn its controls.

Once done, the researcher loads the same exploration technique (basic or advanced) but with the Visualization publications data set discussed in Section 3.2.2. At this point the researcher explicitly states that this is the part of the study that will be recorded by microphone, screen capture and logging of user actions and starts the recordings. Afterward, the researcher hands the Task 1 paper to the user and asks them to verbally answer them as best as possible while freely interacting with the technique, trying out functionalities that appeal to them or seem interesting, and sharing any insights they might have (a “think-aloud” session). Once done, the researcher asks the user to now com-

plete in the same manner as above Task 2. The entire think-aloud session is time-limited to 20 minutes.

With the technique presented and fully explored and the tasks completed, the researcher explains this concludes the recorded part of the study, stops the recordings, terminates the program and presents the follow-up questionnaire to the user. After they fill out the forms, the researcher thanks them again and explains that this concludes the study.

If not from the University collegiate, the user is given 10 EUR for their participation.

5.3 Result Analysis

In total, 10 users participated in this user study, 7 males and 3 females. The minimum age was 25 and the maximum 50, with an average of 32.2 years. 5 of the users were assigned at random to the basic technique and the rest to the advanced one. 4 people reported Bachelor as their highest attained educational degree, 5 answered Master/Diplom and one had a Ph.D. All degrees were attained in Computer Science fields. When asked whether previously or currently employed in an academic field, 4 people gave InfoVis as an answer and 1 responded with “InfoVis/Visual Analytics”.

All but 1 person had some previous experience with data visualizations in general, the rest reporting anywhere between 1 and 15 years, with an average over all participants of 4.6 years. 4 people were already with the vector space model, with 1 additional person answering “a little bit”. However, only 2 people had any experience (2 and 5 years) with the magic lens technique.

Within the allotted time of 15 minutes for the free exploration task all participants were able to eventually identify the core topic of the data set, that being visualization as a scientific endeavor. (Two users, however, immediately

admitted to already being familiar with the data set and having seen or worked with it before.) Even participants without any prior experience with InfoVis were able to quickly identify some application areas like medical or sports visualization.

As can be expected and regardless of which technique was shown, regions with many and more densely packed documents were universally the first target of exploration, which validates the implementation choice of clustering based not on the high-dimensional vectors but on their 2D projections. However, a sentiment shared by several was that the central area was too general and confusing in its topics and avoided it initially, focusing instead on the outlying clusters. This is expected to be an artifact of the t-SNE dimensionality reduction scheme positioning in the central area too many documents that were not very similar to one another.

Several users of the basic technique started out not with a linear search (as was speculated might happen in Section 4.2) but by making the lens as big as possible and trying to get a broader overview of topics. Users of the advanced version, on the other hand, read some of the labels first before interacting with the lens. Both approaches have their merits, but in general users of the latter identified the overall topic or the application areas more quickly.

The second task, which asked users to identify regions that deal with social networking, was also successfully solved by all participants within the 5-minute time limit. Most of them had already passed over one of the two closely grouped clusters that cover the topic during the free exploration phase and acknowledged they no longer remembered where it was. With that said, users of the advanced technique located the static labels positioned over those clusters and solved the tasks very quickly, while the group presented with the basic technique more often had to rely on their memory and a bit of luck to succeed.

The answers to the follow-up questions indicate both groups' generally positive reception to the tool. The functionalities common to both techniques were only marginally more favored by the participants using the advanced visualization. However, the non-common elements – the static labeling, push-out and hiding – usually received very high ratings as well, and basic technique users unanimously agreed that having those features would be beneficial. One user of the advanced application, while agreeing that the static labels were helpful during early exploration, after a few minutes wished he could switch them on/off at will.

Most importantly, several recommendations for further extending the system were proposed by the participants. Beyond the most obvious ones of having tooltips with the document's titles or abstracts available and the integration of a search functionality for filtering, virtually all users requested a feature that would help them trace their steps back during exploration, e.g., by either visually marking already visited regions or with a history slider. One user suggested an improvement to the lens itself – having it automatically resize to always cover the same number number of documents, becoming smaller and more focused in dense clusters and larger and more broad in the sparse regions of the plot. Finally, many participants requested the ability to pin more than one term at once to the lens with AND/OR semantics.

The data gathered by the user logging component of the application can be used to better understand participants' interactions with the technique. An example is shown in Figure 5.1(left), depicting how often a given document was covered by the lens. Despite users often choosing to avoid the central area, they often dragged the lens over it to reach other parts of the visualization. Because that region dominates the values in the plot, a version with a logarithmic scale was created Figure 5.1(right) to better outline differences in coverage.

In summary, the user study confirmed the usefulness of both techniques, with almost universally positive reviews of the newly implemented features. The recorded screen and video recordings, however, confirm that participants presented with the static labeling variant solved both the free exploration and especially the search task quicker.

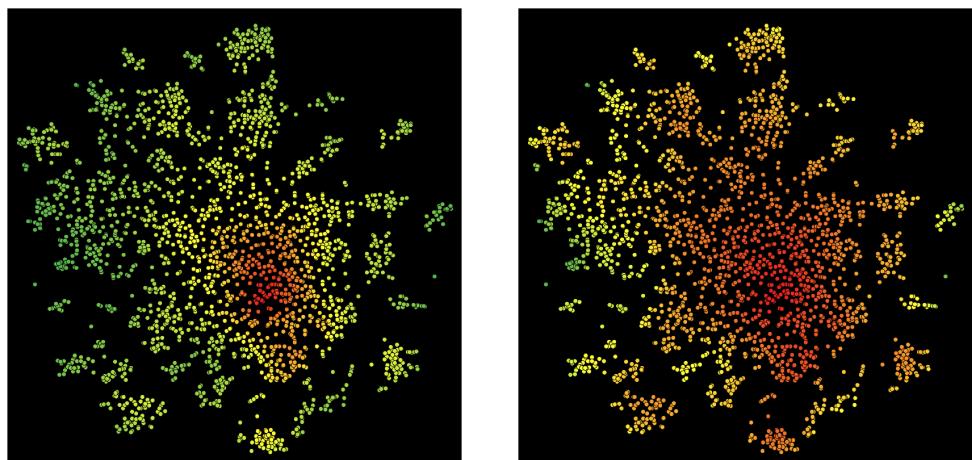


Figure 5.1: Document coverage during the user study. (Left) Number of times a given docupoint was under the lens (with red coloring corresponding to higher values). (Right) The same values on a logarithmic scale. Source: own representation.

Chapter 6: Conclusion

It is becoming increasingly obvious that the sheer volume and the staggering rate of growth of text data produced every single day has long since surpassed the limited processing capabilities of the human brain. However, there also exists an ever-growing body of InfoVis literature and tools that can support users in their exploration of large text corpora and in the sense- and decision-making process. The combined technique of projecting documents as glyphs on a 2D surface and a magic lens to interact with them is a recent contribution to this research.

The goal of this master's thesis was to evaluate and improve on that technique. A review of existing methodologies was necessary to properly position it within its research field and to understand what benefits and drawbacks are inherent in its approach to TextViz. Afterward, a automated framework was conceptualized and developed that evaluates its effectiveness under different parameter settings. Several different means of introducing static, "always-on" labeling to the visualization were developed and their pros and cons were discussed. Finally, a think-aloud user study was conducted that re-affirmed the technique's usefulness as a tool for free exploration of large-scale text data, favorably assessed the newly introduced features and suggested several further improvements.

Appendix A: User Study Consent Form



Lens Evaluation Study Consent Form

Dear participant,

Thank you for agreeing to take part in this study. Please read this form carefully and ask any questions you may have before signing at the bottom.

The purpose of this study is to evaluate a system for exploring large-scale textual datasets. You will first be asked to answer some general questions about yourself, then the capabilities of the system will be explained and demonstrated to you. Afterwards, you will be given a taskset to complete and a follow-up questionnaire to fill out. The entire study will last about 40–45 minutes and will be conducted entirely in English. Please note that during the part of the study that consists of completing the taskset the following data will be gathered: your voice recorded by microphone, a screen recording and a logging of all your interactions with the system, including timestamps.

Your answers to the questionnaires and the data gathered will be confidential. The records of this study will be kept private and will be used only for evaluating the results. Any sort of report made public will not include any information that may make it possible to identify you. You will be given a random number at the beginning to connect your answers with the gathered data, but this number cannot later be traced to a particular person.

Please remember that participation in this study is entirely voluntary. You have the right to ask questions and receive answers at any point. Furthermore, you are free to withdraw from participation at any point; if that is the case, the data gathered so far and your answers to the forms will be destroyed and will not be used in the final evaluation. Finally, you have the right to request short breaks at any time.

Finally, please note that the purpose of this study is to evaluate a visualization system, not you or your personal capabilities. To this end, there are no right or wrong answers, neither to the taskset nor to the questionnaire. You are fully encouraged to give both positive and negative feedback about the system without holding back for fear of insulting the researchers, as your impressions can only serve to improve features that work well and discard those that are not useful.

If you have read the above statements carefully, understand them well and still wish to participate in this study, please explicitly give your consent with your signature below.

Date:

Signature:

Appendix B: Basic Technique Questionnaire



University of Stuttgart, Germany
Institute for Visualization and Interactive Systems

ID: _____

1/5

General Questionnaire

Gender:

- Male
 Female

Age: _____

What is your highest achieved educational degree?

If currently or previously employed in an academic field, please specify which one(s):

How much experience (in years) do you have with data visualization in general?

Are you familiar with the vector-space model for representing text documents?

How much experience (in years) do you have with the “magic lens” technique?

-----► Please stop here for now! ◀-----

APPENDIX B: BASIC TECHNIQUE QUESTIONNAIRE



University of Stuttgart, Germany
Institute for Visualization and Interactive Systems

ID: _____

2/5

Task 1 Please remember to “think aloud”!

What is the overall topic of the dataset? What are major subtopics of the dataset and where are they physically located? Please also talk more about any topics you may be interested in.

-----► Please stop here for now! ◀-----

APPENDIX B: BASIC TECHNIQUE QUESTIONNAIRE



University of Stuttgart, Germany
Institute for Visualization and Interactive Systems

ID: _____

3/5

Task 2 Please remember to “think aloud”!

Find a subset of documents that focuses on social networking. Where are these documents physically located?

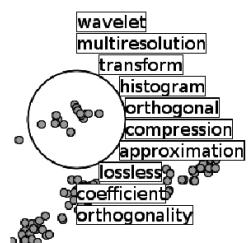
-----► Please stop here for now! ◀-----

APPENDIX B: BASIC TECHNIQUE QUESTIONNAIRE



Follow-up Questionnaire

- 1.** To what degree do you agree or disagree with the following statements? Please select one response per statement.

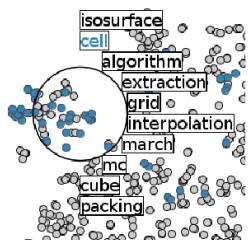


1.1 The lens feature was easy and intuitive to use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.2 The lens feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

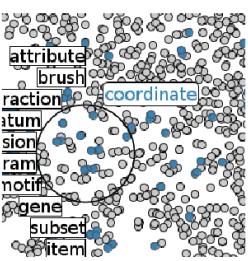


1.3 The term highlight feature was easy and intuitive to use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.4 The term highlight feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



1.5 The term pinning feature was easy and intuitive to use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.6 The term pinning feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

APPENDIX B: BASIC TECHNIQUE QUESTIONNAIRE



1.7 The overall system is easy to interpret and use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Please answer the following questions in the space provided.

1. Do you think the overall system is useful for exploring large textual datasets? Why/Why not?

2. Do you have any suggestions for improvements of specific features and/or the overall system?

3. Do you think that placing static (“always there”) labels over the visualization would have been an improvement of the overall system. Why/Why not?

-----► Thank you for your participation! ◀-----

Appendix C: Advanced Technique Questionnaire



: University of Stuttgart, Germany
: Institute for Visualization and Interactive Systems

ID: _____

1/6

General Questionnaire

Gender:

- Male
 Female

Age: _____

What is your highest achieved educational degree?

If currently or previously employed in an academic field, please specify which one(s):

How much experience (in years) do you have with data visualization in general?

Are you familiar with the vector-space model for representing text documents?

How much experience (in years) do you have with the “magic lens” technique?

-----► Please stop here for now! ◀-----

APPENDIX C: ADVANCED TECHNIQUE QUESTIONNAIRE



University of Stuttgart, Germany
Institute for Visualization and Interactive Systems

ID: _____

2/6

Task 1

Please remember to “think aloud”!

What is the overall topic of the dataset? What are major subtopics of the dataset and where are they physically located? Please also talk more about any topics you may be interested in.

-----► Please stop here for now! ◀-----

APPENDIX C: ADVANCED TECHNIQUE QUESTIONNAIRE



University of Stuttgart, Germany
Institute for Visualization and Interactive Systems

ID: _____

3/6

Task 2 Please remember to “think aloud”!

Find a subset of documents that focuses on social networking. Where are these documents physically located?

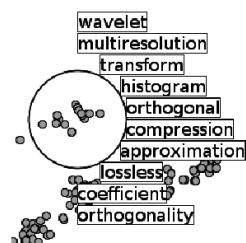
-----► Please stop here for now! ◀-----

APPENDIX C: ADVANCED TECHNIQUE QUESTIONNAIRE



Follow-up Questionnaire

- 1.** To what degree do you agree or disagree with the following statements? Please select one response per statement.

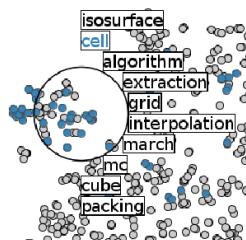


1.1 The lens feature was easy and intuitive to use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.2 The lens feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

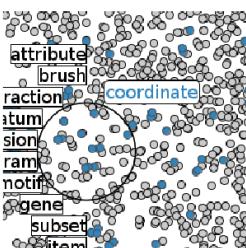


1.3 The term highlight feature was easy and intuitive to use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.4 The term highlight feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



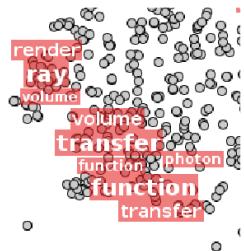
1.5 The term pinning feature was easy and intuitive to use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.6 The term pinning feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

APPENDIX C: ADVANCED TECHNIQUE QUESTIONNAIRE

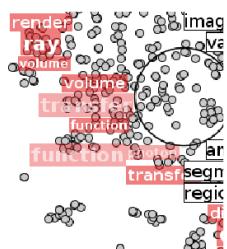


1.7 The static labeling feature was easy to interpret and use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.8 The static labeling feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



1.9 The static labeling push-out feature was easy to interpret.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.10 The static labeling push-out feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



1.11 The static labeling hiding feature was easy to interpret.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.12 The static labeling hiding feature was helpful in solving the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

APPENDIX C: ADVANCED TECHNIQUE QUESTIONNAIRE



1.13 Occlusion by the static labeling was an issue during the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.14 I found the static labeling push-out distracting during the tasks.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.15 The overall system is easy to interpret and use.

Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Please answer the following questions in the space provided.

1. Rank the following features by usefulness while you were completing the tasks:
term hightlight (1), term pinning (2), static labeling (3), static labeling push-out (4),
static labeling hiding (5).

2. Do you think the overall system is useful for exploring large textual datasets? Why/Why not?

3. Do you have any suggestions for improvements of specific features and/or the overall system?

-----> Thank you for your participation! <-----

Bibliography

- [1] M. D. Apperley, I. Tzavaras, and R. Spence, “A bifocal display technique for data presentation,” in *Proceedings of Eurographics*, 1982, vol. 82, pp. 27–43.
- [2] C. Appert, O. Chapuis, and E. Pietriga, “High-precision magnification lenses,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 273–282.
- [3] B. B. Bederson, A. Clamage, M. P. Czerwinski, and G. G. Robertson, “DateLens: A fisheye calendar interface for PDAs,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 11, no. 1, pp. 90–119, 2004.
- [4] E. Bertini, M. Rigamonti, and D. Lalanne, “Extended excentric labeling,” in *Computer Graphics Forum*, 2009, vol. 28, pp. 927–934.
- [5] E. A. Bier, M. C. Stone, K. Fishkin, W. Buxton, and T. Baudel, “A taxonomy of see-through tools,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1994, pp. 358–364.
- [6] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose, “Toolglass and magic lenses: the see-through interface,” in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993, pp. 73–80.
- [7] E. Bier, M. Stone, and K. Pier, “Enhanced illustration using magic lens filters,” *IEEE Computer Graphics and Applications*, vol. 17, no. 6, pp. 62–70, 1997.
- [8] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [9] H. Bosch *et al.*, “Scatterblogs: Geo-spatial document analysis,” in *Proceedings of the 2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2011, pp. 309–310.
- [10] H. Bosch *et al.*, “Scatterblogs2: Real-time monitoring of microblog messages through user-guided filtering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2022–2031, 2013.
- [11] A. Broder, “A taxonomy of web search,” in *ACM Sigir forum*, 2002, vol. 36, pp. 3–10.
- [12] M. Burch, S. Lohmann, F. Beck, N. Rodriguez, L. Di Silvestro, and D. Weiskopf, “Radcloud: Visualizing multiple texts with merged word clouds,” in *Information Visualisation (IV), 2014 18th International Conference on*, 2014, pp. 108–113.

BIBLIOGRAPHY

- [13] M. Burch, S. Lohmann, D. Pompe, and D. Weiskopf, “Prefix tag clouds,” in *Information Visualisation (IV), 2013 17th International Conference*, 2013, pp. 45–50.
- [14] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., 1999.
- [15] Y.-X. Chen, R. Santamaría, A. Butz, and R. Therón, “Tagclusters: Semantic aggregation of collaborative tags beyond tagclouds,” in *International Symposium on Smart Graphics*, 2009, pp. 56–67.
- [16] J. Christensen, J. Marks, and S. Shieber, “An empirical study of algorithms for point-feature label placement,” *ACM Transactions on Graphics (TOG)*, vol. 14, no. 3, pp. 203–232, 1995.
- [17] J. Chuang, C. D. Manning, and J. Heer, “‘Without the clutter of unimportant words’: Descriptive keyphrases for text visualization,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 19, no. 3, p. 19, 2012.
- [18] J. Chuang, D. Ramage, C. Manning, and J. Heer, “Interpretation and trust: Designing model-driven visualizations for text analysis,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 443–452.
- [19] A. Cockburn, A. Karlson, and B. B. Bederson, “A review of overview+detail, zooming, and focus+context interfaces,” *ACM Computing Surveys (CSUR)*, vol. 41, no. 1, p. 2, 2009.
- [20] C. Collins, F. B. Viegas, and M. Wattenberg, “Parallel tag clouds to explore and analyze faceted text corpora,” in *2009 IEEE Symposium on Visual Analytics Science and Technology (VAST)*, 2009, pp. 91–98.
- [21] J. L. Fagan *et al.*, “Method for language-independent text tokenization using a character categorization,” US Patent No. 4,991,094, 05-Feb-1991.
- [22] J.-D. Fekete and C. Plaisant, “Excentric labeling: dynamic neighborhood labeling for data visualization,” in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 1999, pp. 512–519.
- [23] R. A. Finkel and J. L. Bentley, “Quad trees: A data structure for retrieval on composite keys,” *Acta informatica*, vol. 4, no. 1, pp. 1–9, 1974.
- [24] K. Fishkin and M. C. Stone, “Enhanced dynamic queries via movable filters,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995, pp. 415–420.
- [25] C. Forsell, “A guide to scientific evaluation in information visualization,” in *2010 14th International Conference Information Visualisation*, 2010, pp. 162–169.
- [26] G. W. Furnas, *Generalized fisheye views*, vol. 17. ACM, 1986.

BIBLIOGRAPHY

- [27] G. W. Furnas, “A fisheye follow-up: further reflections on focus+context,” in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006, pp. 999–1008.
- [28] M. A. Hearst and D. Rosner, “Tag clouds: Data analysis tool or social signaller?,” in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 2008, pp. 160–160.
- [29] J. Heer, S. K. Card, and J. A. Landay, “Prefuse: a toolkit for interactive information visualization,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2005, pp. 421–430.
- [30] F. Heimerl, Q. Han, S. Koch, and T. Ertl, “CiteRivers: Visual analytics of citation patterns,” *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 190–199, 2016.
- [31] F. Heimerl, M. John, Q. Han, S. Koch, and T. Ertl, “DocuCompass: effective exploration of document landscapes,” presented at the 2016 IEEE Conference on Visual Analytics Science and Technology (VAST), Baltimore, Maryland, USA, 2016.
- [32] F. Heimerl, S. Koch, H. Bosch, and T. Ertl, “Visual classifier training for text document retrieval,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2839–2848, 2012.
- [33] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl, “Word cloud explorer: Text analytics based on word clouds,” in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, 2014, pp. 1833–1842.
- [34] N. Kadmon and E. Shlomi, “A polyfocal projection for statistical surfaces,” *The Cartographic Journal*, vol. 15, no. 1, pp. 36–41, 1978.
- [35] J. Lamping, R. Rao, and P. Pirolli, “A focus+context technique based on hyperbolic geometry for visualizing large hierarchies,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995, pp. 401–408.
- [36] S. Lohmann, F. Heimerl, F. Bopp, M. Burch, and T. Ertl, “ConcentriCloud: Word cloud visualization for multiple text documents,” in *Information Visualisation (iV), 2015 19th International Conference on*, 2015, pp. 114–120.
- [37] J. B. Lovins, “Development of a stemming algorithm,” 1968.
- [38] M. Luboschik, H. Schumann, and H. Cords, “Particle-based labeling: Fast point-feature labeling without obscuring other visual features,” *IEEE transactions on visualization and computer graphics*, vol. 14, no. 6, pp. 1237–1244, 2008.
- [39] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [40] A. M. MacEachren *et al.*, “Senseplace2: Geotwitter analytics support for situational awareness,” in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, 2011, pp. 181–190.

BIBLIOGRAPHY

- [41] J. D. Mackinlay, G. G. Robertson, and S. K. Card, “The perspective wall: Detail and context smoothly integrated,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1991, pp. 173–176.
- [42] C. D. Manning, P. Raghavan, and H. Schütze, *An introduction to information retrieval*. Cambridge University Press, 2008.
- [43] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP natural language processing toolkit,” in *ACL (System Demonstrations)*, 2014, pp. 55–60.
- [44] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou, “TreeJuxtaposer: scalable tree comparison using focus+context with guaranteed visibility,” in *ACM Transactions on Graphics (TOG)*, 2003, vol. 22, pp. 453–462.
- [45] A. Nenkova and K. McKeown, “A survey of text summarization techniques,” in *Mining text data*, Springer, 2012, pp. 43–76.
- [46] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz, “Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 564–575, 2008.
- [47] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.
- [48] R. Rao and S. K. Card, “The table lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1994, pp. 318–322.
- [49] A. W. Rivadeneira, D. M. Gruen, M. J. Muller, and D. R. Millen, “Getting our head in the clouds: toward evaluation studies of tagclouds,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, pp. 995–998.
- [50] G. G. Robertson and J. D. Mackinlay, “The document lens,” in *Proceedings of the 6th annual ACM symposium on User interface software and technology*, 1993, pp. 101–108.
- [51] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.
- [52] M. Sarkar and M. H. Brown, “Graphical fisheye views of graphs,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1992, pp. 83–91.
- [53] B. Schneiderman, “The eyes have it: A task by data type taxonomy for information visualizations,” in *Proceedings of the 1996 IEEE Symposium on Visual Languages*, 1996, pp. 336–343.

BIBLIOGRAPHY

- [54] M. C. Stone, K. Fishkin, and E. A. Bier, “The movable filter as a user interface tool,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1994, pp. 306–312.
- [55] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann, “A survey on interactive lenses in visualization,” *EuroVis State-of-the-Art Reports*, vol. 3, 2014.
- [56] J. Viega, M. J. Conway, G. Williams, and R. Pausch, “3D magic lenses,” in *Proceedings of the 9th annual ACM symposium on User interface software and technology*, 1996, pp. 51–58.
- [57] F. B. Viégas and M. Wattenberg, “TIMEELINES: Tag clouds and the case for vernacular visualization,” *Interactions*, vol. 15, no. 4, pp. 49–52, 2008.
- [58] F. B. Viegas, M. Wattenberg, and J. Feinberg, “Participatory visualization with wordle,” *IEEE transactions on visualization and computer graphics*, vol. 15, no. 6, 2009.
- [59] J. A. Wise, “The ecological approach to text visualization,” *Journal of the Association for Information Science and Technology*, vol. 50, no. 13, p. 1224, 1999.
- [60] J. A. Wise *et al.*, “Visualizing the non-visual: Spatial analysis and interaction with information from text documents,” in *Information Visualization, 1995. Proceedings.*, 1995, pp. 51–58.
- [61] K. Yatani, M. Novati, A. Trusty, and K. N. Truong, “Review spotlight: a user interface for summarizing user-generated reviews using adjective-noun word pairs,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 1541–1550.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.

Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Unterschrift:

Stuttgart, 24.1.2017

Declaration

I hereby declare that the work presented in this thesis is entirely my own.

I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations.

Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before.

The electronic copy is consistent with all submitted copies.

Signature:

Stuttgart, 24.1.2017