VarifocalReader – In-Depth Visual Analysis of Large Text Documents

Steffen Koch, *Member, IEEE*, Markus John, Michael Wörner, Andreas Müller, and Thomas Ertl, *Member, IEEE*

Abstract— Interactive visualization provides valuable support for exploring, analyzing, and understanding textual documents. Certain tasks, however, require that insights derived from visual abstractions are verified by a human expert perusing the source text. So far, this problem is typically solved by offering overview+detail techniques, which present different views with different levels of abstractions. This often leads to problems with visual continuity. Focus+context techniques, on the other hand, succeed in accentuating interesting subsections of large text documents but are normally not suited for integrating visual abstractions.
With VarifocalReader we present a technique that helps to solve some of these approaches' problems by combining characteristics from both. In particular, our method simplifies working with large and potentially complex text documents by simultaneously offering abstract representations of varying detail, based on the inherent structure of the document, and access to the text itself. In addition, VarifocalReader supports intra-document exploration through advanced navigation concepts and facilitates visual analysis tasks. The approach enables users to apply machine learning techniques and search mechanisms as well as to assess and adapt these techniques. This helps to extract entities, concepts and other artifacts from texts. In combination with the automatic generation of intermediate text levels through topic segmentation for thematic orientation, users can test hypotheses or develop interesting new research questions. To illustrate the advantages of our approach, we provide usage examples from literature studies.

Index Terms—visual analytics, document analysis, literary analysis, natural language processing, text mining, machine learning, distant reading

1 Introduction

Visual abstraction of text documents can support users in getting a general understanding of the information a text conveys and in judging its relevance without having to actually read through it. This can be very helpful when an analysis task requires the user to consider large volumes of complex text. However, working in the abstract is often not enough to solve the entire task, and access to the text level is required to verify and prove findings or hypotheses. This has been acknowledged by many visual analytics (VA) approaches for text analysis tasks, which commonly offer overview+detail methods to switch between an abstract visual representation and the text level. Well-known approaches have been described, for example, by Wise et al. [47] for creating a spatialization of documents in order to help users understanding similarities of documents, by Stasko et. al. [38] for extracting named entities and their assumed relationships from textual documents, and by Oelke et al. [27] for assessing a document's readability based on a variety of text characteristics. VA tasks with their changing information needs, which typically form and evolve during iterative analysis [40, 41, 20], can become intricate if the context can only be switched from high aggregation levels to the text level directly.

With VarifocalReader, we aim at smoothing out these context switches for the analysis of single, large text documents by introducing additional intermediate levels of abstraction that can be navigated and explored interactively. These intermediate levels can be derived from a document's intrinsic logical or layout-based structure. If a document's formal structure is either too coarse or too fine-grained, the user can ask for additional levels to be generated by automatic topic segmentation. The hierarchical perspective on the text document and the mechanisms for exploring and navigating it are based on the Smooth-Scroll [48] concept, which shows aspects of both overview+detail and focus+context interaction.

As with other visual analytics tasks, visual text analysis can profit from a combination and integration of different automatic methods and interactive visual techniques. When dealing with textual data, natural language processing (NLP) techniques are unsurprisingly the methods of choice for automatic analysis. VarifocalReader includes some NLP techniques and can be flexibly extended with additional methods to support human analysts. Moreover, its hierarchical interface supports automatic analysis methods because their effects can be tracked easily by visualizing them in hierarchical layers of different levels of abstraction. Besides topic segmentation, our approach offers named entity recognition (NER) and automatic summarization of text segments through word clouds to support navigation and exploration tasks within a text document. Furthermore, task-tailored NLP and machine learning can be easily added to the modular approach, as can visual abstractions for conveying results to users. An example for exploiting a machine-learning based classification approach in VarifocalReader is provided in Section 4.

In recent years, many NLP techniques have matured and display impressive robustness. They are typically based on statistical approaches and machine learning. This results in methods that work best on common, widely electronically available types of texts since they were optimized, trained, and tested on similar resources. Consequently, NLP methods may be less effective when applied either to very specific tasks or to particular kinds of documents, such as historical texts. Because large training sets are most often not available in these situations, it can be more effective to let users adapt techniques available out of the box. The VarifocalReader approach has been designed to let users perform such adaptations on a per document level.

VarifocalReader has been developed in cooperation with literary scholars, natural language processing specialists, and computer philologists as part of the digital humanities project 'ePoetics'. Its primary goal is to enhance the support for analysis tasks that are part of the research efforts of literary scholars. In the study of literature, the text document itself is the primary object of research. Visualization can be applied to help with this research by illustrating overarching, summarizing textual aspects in order to help scholars develop new research ideas, form hypotheses, etc. – an idea that is not new to scientists from the humanities (cf. Moretti [24]). However, literary research has to be done on the text itself or at least be verified by the actual docu-

- *Andreas Müller is with the Institute for Natural Language Processing (IMS), University of Stuttgart, e-mail: Andreas.Mueller@ims.uni-stuttgart.de*
- *All other authors are with the Institute of Visualization and Interactive Systems (VIS), University of Stuttgart, e-mail: <firstname.lastname>@vis.uni-stuttgart.de*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014. Date of publication 11 Aug. 2014; date of current version 9 Nov. 2014.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.
Digital Object Identifier 10.1109/TVCG.2014.2346677

ment, which led to the decision to develop VarifocalReader with direct access from high-level visualization to the text level.

The ePoetics project aims at analyzing a rather specific kind of literature, namely German poetics [33] dating from 1770 to 1960. Poetics are meta-literature discussing, amongst other things, the history, aesthetics, and criticism of literary texts and can be regarded as one of the building blocks of modern literary studies. Unlike many other visual text analysis tasks, this use case requires that we deal with few but large and complex documents. The problem therefore is not the amount of textual data to be analyzed but the quality, detail, and task-specificity that is necessary in order to gain new research insights from the text analysis. The humanities are, by definition, a field that relies on human intuition and steering. Nevertheless, they can benefit from the inclusion of automatic methods and graduated visual abstraction and appear to be a promising application domain for visual analytics techniques. We believe that other tasks, where scrutiny of analyses on the text level is mandatory, can also profit from developing such approaches.

In the context of ePoetics, VarifocalReader is applied for different purposes. Here, researchers are interested, for example, in answering rather abstract questions, such as "how does the author of a poetic describe the concept *poet*" and "which facets of a poet are described". Carrying out a search for "poet" or applying more sophisticated methods to extract this concept, it is interesting to see that different authors of German poetics characterize poets according to the type and style of their works. For example, Schiller's "Wallenstein" is often cited in the context of dramas, raising the question of whether Schiller's works in general are regarded as prototypes of dramatic poetry or only "Wallenstein" is considered to be prototypical. This can quickly lead to follow-up questions, e.g., which other works are referenced in this context. In order to work with such citations of dramatic poetry and to use them as robust data for further analyses, literary scholars might be interested in annotating such citations explicitly after verifying them in the text itself.

Situations like the one described in the example above led to the development of VarifocalReader. In particular, the combination of extraction mechanisms with exploratory analysis – including the embedding of findings in summarized contexts, the verification of findings on the text document level, and their explicit representation through annotation – turned out to be a very useful combination. The questions and goals of literary scholars in the ePoetics project are manifold: from developing abstract research questions to testing hypotheses to explicit high-quality annotation of literature. Therefore, VarifocalReader was designed to let scholars perform a variety of analysis tasks of diverse abstraction, varying granularity, and different requirements for automatic analysis quality in one interactive visual tool.

The overall goal of the ePoetics project is the development of interactive visual analysis methods combined with techniques from corpus linguistics in order to advance computational methods with hermeneutic methods of the humanities in the sense of an "algorithmic criticism" [30]. The VarifocalReader approach is a first step in this direction.

The remainder of this work is structured as follows: Section 2 gives an overview of related work. Our approach is described in detail in Section 3. Subsequently, a usage scenario, including analysis tasks that can be carried out, is presented in Section 4. Section 5 discusses the approach and the feedback we received from our expert before we conclude the paper in Section 6.

2 Related Work

VarifocalReader relates to previous work under three different aspects. Firstly, techniques for visualization and visual interfaces for navigating, browsing, and exploring text documents are relevant in the context of our work, which relies on a scrolling technique called SmoothScroll [48]. Secondly, several existing approaches for the visual abstraction of textual content are discussed below, since abstractions and summarizations are an integral part of our technique as well. Finally, this section looks at visual analytics approaches that incorporate advanced analysis methods, such as machine learning, which can be applied and adapted by users through interactive visualization.

2.1 Document Navigation and Exploration

Nowadays, exploring and browsing electronic documents containing considerable portions of text is probably one of the most common tasks carried out when working with computers. It is therefore not surprising that many ideas on navigating and browsing electronic text documents have been suggested. Cockburn et al. summarize and categorize interaction techniques relevant for browsing text documents in [7]. The technique we present exhibits characteristics of overview+detail, focus+context, and cue-based methods if using the categorization of Cockburn et al. Scrolling techniques are amongst the most commonly used techniques for realizing overview+detail interaction.

The approach we use is based on SmoothScroll by Wörner and Ertl [48]. SmoothScroll was originally developed to support browsing large amounts of hierarchically structured data. Its main idea is to show a low-detail view of the entire data set at the coarsest level of the hierarchy while distributing subordinate levels using hyperbolic distortion, which show only a portion of these levels at a time. The closer a level is to the leaf-level of the hierarchy, the stronger is the distortion and the more space is available for showing details. With respect to the applied distortion and showing full context on the coarsest level, SmoothScroll shares some characteristics with common focus+context techniques such as fisheye views [15]. At the same time, it separates information in different hierarchical levels without continuous transitions, which is typical for overview+detail methods.

The SmoothScroll approach is especially well-suited in situations where the underlying data is sequential in nature because the technique always shows 'neighboring' data items of a focused region on each hierarchical level but hides items that are sequentially far away from the current focus. SmoothScroll uses a space-filling approach that is similar to icicle plots [22]. Its main difference to icicle plots is that it relaxes the visual relation between the items on different levels, which is determined as each parent icicle spanning exactly all child icicles. Visual child-parent assignment therefore has to be resolved by other means. SmothScroll can be seen as an interactive, hyperbolic variant of icicle plots. Its primary area of application has so far been the display of time-dependent data, but its potential application to text data was already anticipated in [48].

VarifocalReader incorporates SmoothScroll's basic idea but extends and advances it in several ways to improve working with large text documents. The 'Document Lens', which was presented by Robertson and Mackinlay [34], is a focus+context technique that applies geometrical distortion for browsing documents with unknown structure, while VarifocalReader distorts hierarchical document structures discretely. In this respect, our approach exhibits more similarities to the 'Table Lens' [31] and its extension for multiple focal levels [39], even though our approach is not bifocal and distortion is constantly available through the different hierarchical levels. A detailed discussion of how VarifocalReader can be used to navigate and explore text is provided in Section 3.

2.2 Visual Text Abstraction

Several techniques have been developed for visually abstracting or summarizing text documents, depicting specific text characteristics, and highlighting search results. These techniques were developed for, often domain-specific, text analysis tasks, e.g, the depiction of (web) search results with Tilebars [17], summarization of source code for visualizing software statistics (Seesoft [13]), or, as in the digital humanities, for literature analysis tasks [21].

Besides using a line-based summarization technique for depicting text characteristics similar to Seesoft, VarifocalReader provides bar charts for summarizing occurrences of text characteristics and word clouds for summarizing text content as a starting point for further analysis [44].

Of course, this is not a complete set of possible visualization techniques that could be used for depicting text or certain characteristics

of texts. As soon as structures have been derived or text can be represented by a model, almost all common techniques for displaying such information or its aggregation are available to support analysis tasks [35].

To present relational information extracted from textual resources, visualization techniques such as WordTrees [46] and PhraseNets [43] were proposed. With DocuBurst, Collins et al. [8] suggested a method for representing document contents by depicting relevant terms with a space-filling approach that acknowledges semantic relationships of these terms. The mentioned approaches abstract relationships of text properties visually but do not integrate access to the source text for detailed analysis.

2.3 Visual Text Analytics

During the last years, more and more VA approaches for analyzing text have been suggested [47, 38, 27] to support users in understanding document similarity, extracting named entities, and infering their relationships. Additionally, systems for assessing the readability of texts [28], for extracting sentiment information from customer reviews [25], as well as techniques for extracting events, trends, and topics from social media data were proposed [11].

There are also quite a number of approaches that particularly support visual analysis for the digital humanities and more specifically for literary sciences. Corell et al. [9] provide an interactive visual framework for the analysis of large literary text collections that have been tagged by NLP preprocessing. It supports user-steered separation and exploration of text corpora based on principle component analysis. A work on rule-based solutions for poetry visualization is offered by Abdul-Rahman et al. in [1]. Recent work by Oelke et al. [26] discusses the visual analysis of prose literature through 'Fingerprint Matrices', a technique for analyzing implicit relationships of characters. Most relevant to our approach is the tool called POSvis, which was presented by Vuillemot et al. in [45]. Similar to the work of Oelke et al. POSvis aims at exploring named entities in literary texts. POSvis makes use of a multiple coordinated views approach, giving different perspectives. Our approach differs insofar as it offers a finer granularity of hierarchical segmentation and lets users apply and adapt NLP and mining methods.

VarifocalReader offers topic segmentation to subdivide parts of texts according to the user's needs. For this purpose, we employ the C99 algorithm as suggested by Choi in [6]. Topic segmentation is different from topic extraction [4], which has been suggested as part of visual analysis techniques for summarizing multiple documents and was addressed by several recent works [23, 12]. Other techniques for relating topics in document collections exist, as, e.g., described in [32].

Approaches have been presented, e.g. [10], that incorporate visualization and automatic methods to improve text analysis tasks. As of yet, however, very few approaches are available that enable users to influence or adapt the automatic analysis method in order to achieve higher quality during text mining tasks. For the field of literature analysis, achieving this higher quality level can be a requirement. If this is not possible with semi-automatic methods, it has to be done manually to generate reliable results. Nevertheless, automatic methods can help to detect certain trends or to form hypotheses. They are even more useful if they can be adapted to the type of text and the research task. Approaches offering interactive visual methods for user-steered adaptation of analysis methods are still rare. Brown et al. [5] suggested a method to develop similarity functions for machine learning models in a visual and interactive way, liberating them from understanding the complex parametrization of those models. Similarly, Hu et al. [19] describe interaction methods for parameter adaptation which they call "semantic interaction". Here the parametrization of models is adapted implicitly by user interactions on spatializations of these models. Endert et al. [14] have proposed this form of semantic interaction for document analysis techniques. Their method exhibits similar properties in terms of letting users directly label examples for subsequent training of machine learning techniques. An approach that lets analysts interactively train a support vector machine for text document classification based on a visual explanation has been presented by Heimerl et al. [18].

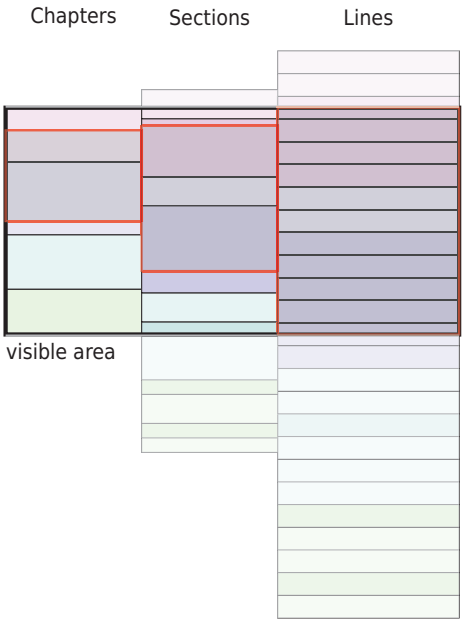

Fig. 1. Schematic representation of a three-layer SmoothScroll view. The left layer displays a coarse view of the entire text document, the right layer a detailed but clipped view of individual lines. Highlights indicate which portion of the less detailed layers correspond to the section visible on the detail layer.

3 Approach

The VarifocalReader approach supports users in exploring and understanding complex text documents by visualizing them at various aggregation levels. It provides users with means for navigating the visualization, tracing the current position across all aggregation levels, and offers automatic support from text processing and machine learning algorithms.

The various tasks and goals of humanities scholars when analyzing text resources are comparable to sensemaking processes as described by Pirolli and Card [29]. Of course, there are differences as well: In our approach, information foraging focuses on single documents. This led to the design decision to create a technique that facilitates parallel usage of close and distant reading, or at least the means for switching very quickly between these modes realized by the hierarchical browsing approach. Different visual views can then be used on these hierarchies to display the results of search, machine learning, and manual annotation in a context adequate to the users' needs. This enables users to flexibly drill down into details, e.g., to detect and find interesting text characteristics. Additionally, they can immediately test whether these features are just available locally or represent systematic global properties. VarifocalReader provides different mechanisms for information foraging, such as search and other extraction mechanisms. It supports sensemaking by letting users relate a finding to another on different levels of abstractions and in different contexts, and, most importantly, besides this fast context switching and the quick changing between different views, it offers users various feedback-loops in order to let them react to new findings, to test hypotheses, to improve extraction mechanisms, and to observe effects on different document scales.

3.1 Multi-layer visualization and navigation

VarifocalReader displays text documents in a multi-layered view akin to the SmoothScroll control described in [48]. SmoothScroll provides a way of navigating a sequential data set, i.e., a set of data items or-

dered along one dimension such as time. Our data sets are text documents, which are essentially sequences of symbols or characters. The available screen space is divided into multiple layers that display portions of the data set at different levels of scale and abstraction (Figure 1). The most detailed layer displays individual data items, which in our case are individual lines of text. We refer to this layer as the 'detail layer'. The least detailed layer displays the entire range of the data set, which will usually make it infeasible to draw single data items and require appropriate aggregation. We call this the 'overview layer'. The layers in between interpolate their scale between these two extremes and may use various intermediate aggregation levels.

When a user scrolls through the data set, the focus position, i.e., the line of text at the center of the detail layer, is synchronized across all layers. A highlight on the other layers indicates the location and extent of the section visible on the detail layer. Except for the fixed overview layer, all layers move to keep this section in view. The user can use any layer to scroll or jump to a position and can thus make very fine-grained changes on the detail layer or cover large distances on one of the less detailed layers. The navigation is straightforward: right-clicking any point will set the focus to this position and adjust all layers accordingly. It is also possible to move the mouse while the right mouse button or the mouse wheel is pressed, continuously browsing through the data. Because of the different scales, the speed at which the focus point moves changes as the mouse pointer passes over different layers and the user can influence the speed of the movement by moving the mouse pointer to another layer while scrolling through the data. The detail layer moves quickly, whereas the layers to the left of it move gradually slower.

One advantage of using several discrete layers instead of a continuous distortion is that each layer can display data items at a different level of abstraction or aggregation. Many text documents, for example, are logically structured into chapters, sections, subsections, or similar divisions, so a SmoothScroll view of a text document may list the chapters of the text on the least detailed layer, relate the current position to sections and subsections on the intermediate layers, and display the individual lines of text on the detail layer. This visualization lets the user read the actual text at the focus position and at the same time retain a sense of the 'big picture'. Our data sets are digital representations of physical books and thus also have a physical structure in terms of lines and pages. These subdivisions are relevant whenever analysts need to refer to the original work. The inherent logical and physical structure of a text document can be augmented with the results of text analysis methods such as topic segmentation and mostly manual analysis steps such as term highlighting. Theoretically, topic segmentation makes it possible to generate a structured document from one that has no initial partitioning at all, as long as it offers continuous text.

3.2 Automatic analysis

VarifocalReader offers a certain amount of automatic support for analyzing natural language text. It can generate word clouds and topic segmentations, draw bar charts showing the number of occurrences of annotations or search terms, display pictograms that relate the location of highlighted text segments to the physical layout of the original text, and show labels that list headings, page numbers, or the content of a line of text.

A **topic segmentation** algorithm can estimate where a section of text deals with a certain topic and where the predominant topic changes. The C99 algorithm we use judges the similarity of text blocks by the cosine distance between their term frequency vectors after lemmatization and stop word removal. We chose C99 because it is very fast compared to other topic segmentation algorithms [6]. Our main motivation for using topic segmentation is that we want to provide an additional level of document structure that lists the predominant topics of a document. We perform the segmentation on the chapter and subchapter level and with sentence granularity rather than on single pages or paragraphs. The reason we do not perform topic segmentation, e.g., on the page or paragraph level, is twofold: Hierarchical partitions that result from layout rather than structure often do not describe "complete" or "meaningful" topics at their beginning or at the end, since in these places sentences are likely to be split. One could think about completing these parts to full sentences or even to paragraphs, but this would break with our design decision to depict only aspects that are contained in a partition. For typically short paragraphs, reading them might give better insight into the topic than a rather high-level summarization with key terms. On the other hand, chapters and subchapters normally contain multiple topics. Also, topic segments can span more than one page or paragraph. Therefore, segmenting them would lead to incomplete topic segments.

In general, applying topic segmentation is particularly useful for large documents of several hundred pages, such as books, as in our example, or very large documents (e.g., court files), as long as they represent continuous text. Topic segmentation is less useful for shorter texts such as news articles and blog entries.

Word clouds generated for a body of text can serve as a starting point for deeper analysis. We use the part-of-speech (POS) information from NLP preprocessing to reduce the text of a segment to the contained nouns, since these are likely to reflect the discussed concepts. We are currently using the POS tagger from the mate-tools (`https://code.google.com/p/mate-tools/`) for German texts and the tagger from the Stanford CoreNLP library [37] for English ones. For computing the word clouds, we use the standard scoring formula of Lucene [2]. This formula is a variant of the basic TF-IDF scoring often employed in information retrieval. TF-IDF considers not only the frequency of a word within the text section represented by a layer item (its term frequency – tf) but also in how many other items on the same layer the word appears (in terms of its inverse document frequency – idf). As a result, even if a word appears only once within the scope of a layer item, it can still be a member of the item's word cloud if it does not appear in any other item of the same layer, making the words in the cloud representative of the corresponding text section.

Rule-based and machine learning-based **automatic annotation** can, for example, highlight all literal quotations or proper names in a text. We use the Stanford CoreNLP tools [37] for this purpose with German and standard English models to detect the names of persons and locations. For detecting quotes in text, we use the regular expression pattern `".*?"` making each text enclosed in quotation marks a quote. The lazy quantifier `*?` is required to prevent text between the `"` at the end of one quotation and the `"` at the start of the next quotation to be incorrectly marked as a quote.

An **active learning** algorithm can be used to create automatic annotations for concepts that are more complex than what can satisfactorily be expressed by rule-based annotation. For example, when applying the quotation rule to Emil Staiger's poetic "Grundbegriffe der Poetik" ("Basic Terms of Poetics", [36]), we noticed that text inside quotation marks does not always constitute a quotation from another work. Rather, Staiger also uses quotation marks to mark titles of other works or to emphasize certain words that are important in the current context but neither quotations nor titles. We addressed this by integrating the support vector machine classifier provided by the Weka project [16]. We use either the user's manual annotations or a set of rule-based annotations as an initial set of instances on which we train the classifier. Each text between quotation marks is represented by a small number of surface features. An example for one such feature is whether the text between quotation marks is preceded by a person's name. In this case the text between quotation marks is probably a quotation rather than an emphasis or title. We then apply this classifier to the entire text to create appropriate annotations. The Weka classifiers also return a confidence value, which we retain and communicate to the user. Note that, in principle, we could re-train every part of the pre-processing pipeline. The re-training of the quotation classifier is just an example for how the re-training mechanism works in general. Re-training components like the POS tagger is much more complex, which is why we chose to implement the first version of the re-training mechanism for a simple classification problem. The output of the quotation detection is directly relevant for the literature scientists.

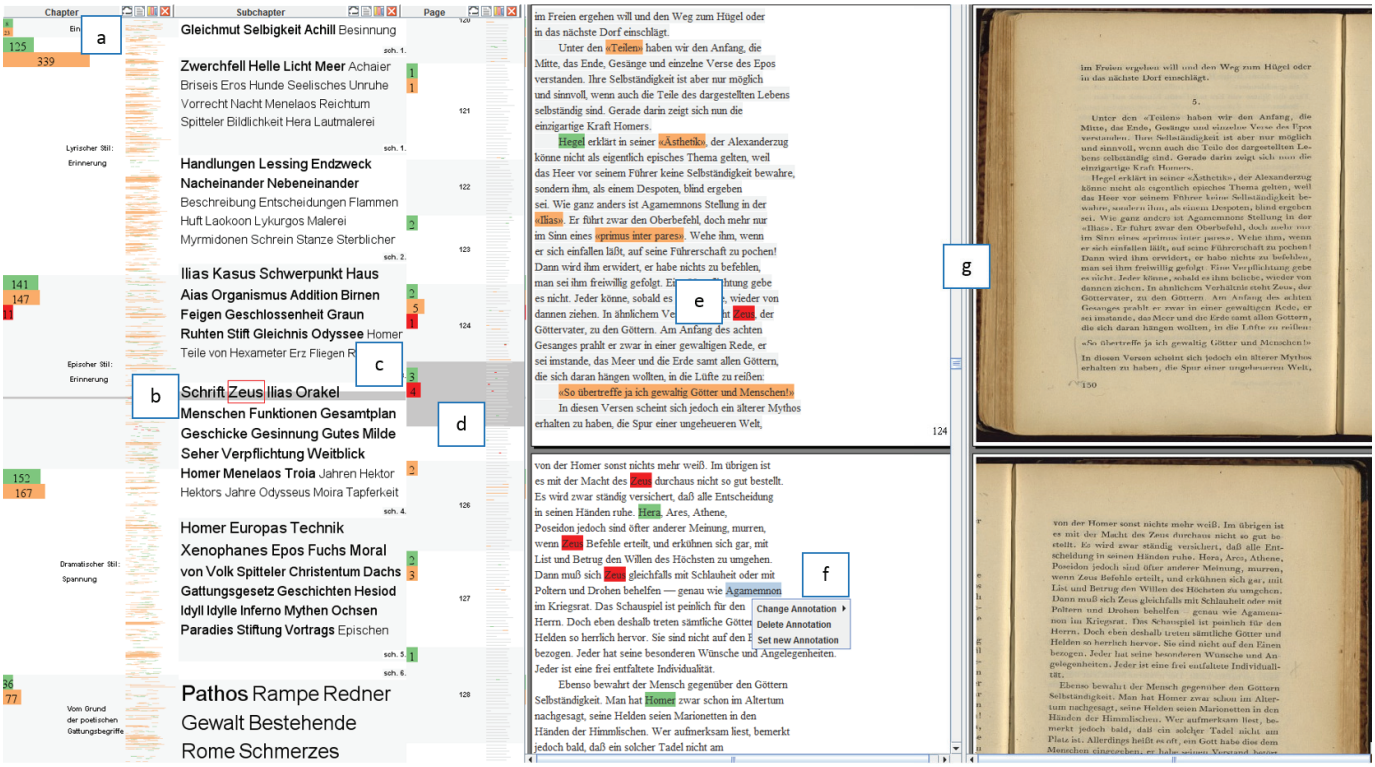

Fig. 2. Emil Staiger's "Grundbegriffe der Poetik" divided (from left to right) into layers showing chapters, subchapters (with word clouds), pages (with bar charts), lines of text, and scanned images of the actual pages.

3.3 VarifocalReader

Figure 2 depicts the main workspace of our VarifocalReader after loading an exemplary literature document. Little icons in the layer headers (Figure 2a) provide the option to enable on-layer displays of word clouds, bar charts, and pictograms. It is also possible to hide a particular layer from view. Additionally, users can resize layers horizontally by dragging the column header borders. This makes the VarifocalReader approach very flexible because different aspects of the document can be viewed simultaneously. Analyst can therefore focus on those aspects of a document that are most relevant to their current task, without being distracted by non-essential information.

VarifocalReader uses **word clouds** to give a visually appealing overview of a section of text. This summarization is useful for learning about the number and kind of topics present in a body of text. We organize these word clouds in a sequential layout with horizontal and vertical arrangement of tags sorted by the score returned by a Lucene query for the respective word (Figure 2b). Additionally, the user can search the text section corresponding to the current layer item for an individual word by clicking on it (Figure 2b). The search results are highlighted on the different layers in red.

Bar charts show the number of occurrences of annotations or search terms in the respective item (Figure 2c), and **pictograms** display the location of highlighted text segments (Figure 2d). Annotations and search terms are also highlighted on the layer showing individual lines (Figure 2e). In case annotations overlap in the text, we draw them using semi-transparency and alpha blending. A user can select words or existing annotation and delete, change, or add new annotations using a context menu (Figure 2f).

While it is important to be able to visualize the distribution of term occurrences in documents, statistics alone do not show all aspects of these occurrences and may hide significant details. Therefore, we highlight all affected annotations on all the layers when the user selects an annotation or bar chart (Figure 3). This makes it possible to consider the annotations within their context in the actual text and is important in order to assess the significance of the statistical results. For example, a user can see whether the occurrences are far apart or concentrated in few regions of text.

Fig. 3. The annotations contributing to the value of the selected bar chart on the second layer are highlighted in red on the detail layer.

VarifocalReader can also display scanned images of the original pages next to the detail layer (Figure 2g). This gives immediate access to all nontextual information within a source document whenever the need arises. Nontextual information may include pictures or handwritten text, which will usually look different when converted to a digital text format.

Topic segmentation is especially useful when a document lacks the formal structure expected from long documents or when its structural elements do not give any indication of what the element is about. It can also be useful when the structural elements are very long and without further subdivisions. In those cases, the automatic division of long structural elements can help users to find discussions about topics of interest more quickly. For example, Staiger's chapter "Vom Grund der poetischen Gattungsbegriffe" ("On the Foundation of Generic Terms in Poetics") contains many topics like Pflanze (plant), Geist (spirit), Sprache (language), or Zeit (time), as can be seen in the left part of Figure 4. The topic segmentation algorithm divides this chapter into six segments (visible in the right part of Figure 4). It is apparent that

these segments focus on different aspects of the chapter. For example, the first segment talks about Pflanze (plant) and Tierwelt (animal domain) while the following segments discuss topics like Seele (soul) and Geist (spirit) or temporal aspects like Gegenwart (present) and Vergangenes (past things). This overview lets users find potentially interesting topics by examining the word clouds for related words. In effect, the topic segmentation layer elaborates on the word cloud summary of the chapter layer by adding information on the sequence and combination in which these key terms appear. If an analyst intends to examine a specific topic, these segments provide an indication of which parts of the chapter are most relevant.

Fig. 4. The word clouds for a chapter (left) and the chapter's automatically generated topic segments (right) of Staiger's book [36].

Another example for the utility of topic segmentation of a chapter is depicted in Figure 5. This Figure shows, on the left, the word cloud for chapter 9 of the Iliad. On the right we see the word clouds for the topic segments computed for this chapter. Because the Iliad tells a story, word clouds have to be interpreted differently than in Staiger's poetic. The word clouds in Figure 5 give an overview of the types of events the chapter consists of. In the first word cloud, we see a theme expressed by the related terms "ambassadors, judgment, scepters, king, council and wisdom". In the next word clouds, we see the central entities appearing in the story, e.g., the two characters "Patroclus" and "Ulysses" in the second cloud, and "Althaea" and "Phoenix" in the fourth. The word clouds do not denote abstract topics as in the poetic of Staiger but give an indication of the parts of the story told in a chapter.

Annotations resulting from the active learning component include a confidence value, which is visualized in the opacity of the annotation highlight. Faint highlights indicate uncertain annotations, which are good candidates for manual annotation. The user can correct or confirm these classifications and trigger a retraining to improve the classifier by considering both the original training set and the additional information. Once retraining is complete, both the old and the new classification is shown to enable users to quickly assess the success of the retraining. Figure 6 shows a case in which quoting the term "Antigone" was incorrectly classified as an emphasis (green) when in fact it is a reference to the classical Greek tragedy and should therefore be classified as a title (purple). This error can be corrected using a context menu. Manual correction and retraining can be repeated as often as necessary and greatly reduces the amount of training data needed to produce a high-quality classifier.

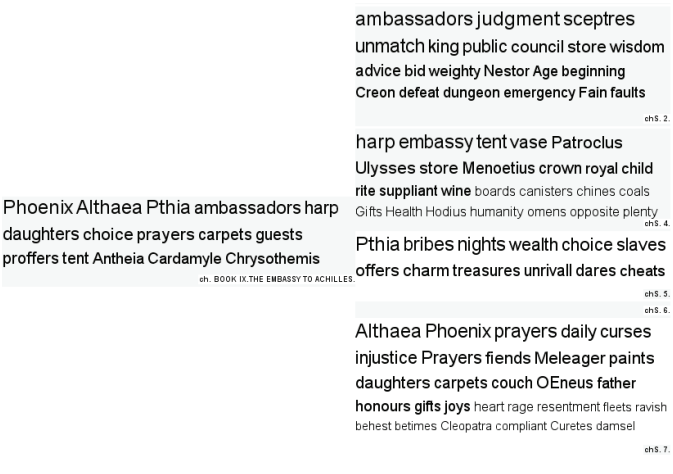

Fig. 5. The word clouds for chapter nine (left) and the chapter's automatically generated topic segments (right) of the Iliad by Homer

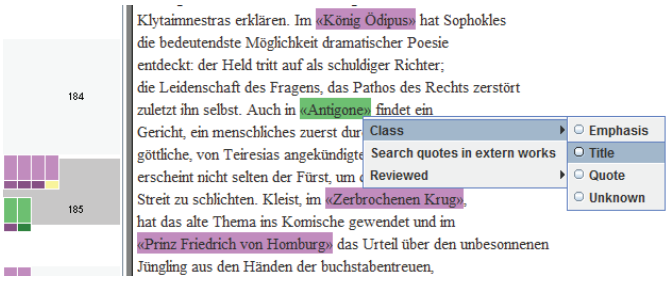

Fig. 6. The incorrect classification of "Antigone" as an emphasis can be corrected manually to improve the automatic classification. The effects of the previous training step are visible in the rectangular glyphs on the left. The fourth occurrence of an annotation (upper row of glyphs), which is currently classified as an emphasis (purple), was classified as "unknown" before as indicated by the yellow bar. The wrongly changed label/classification of "Antigone" is depicted by the first green (title) glyph with the purple (emphasis) bar below it in the second row. Glyphs are always sorted by current class assignment.

VarifocalReader also supports loading a second document to compare two texts side by side. For this scenario, the view is duplicated and mirrored to the right of the original one (Figure 7). This can be used, e.g., to compare term distributions in two similar documents or predominant terms at similar positions in the documents to discover similarities and differences between the documents. It is also useful for comparing documents that discuss similar topics from different points of view because topic segmentations and word clouds facilitate finding the sections relevant for a topic in either document. In a similar way, different editions of the same work can be compared in order to identify relevant changes more quickly.

4 Usage Scenario

In the following, we present a usage scenario that demonstrates the suitability of VarifocalReader for analysis tasks. Additionally, we describe a use case that shows how an English version of Homer's "Iliad" can be analyzed with our approach. The analysis of the first use case is derived from real research (sub)tasks suggested by our project partners from the institute of German literature studies. The second use case illustrates the applicability of our approach to English documents.

As previously mentioned, the ePoetics project deals with the analysis of German poetics, which can be seen as the predecessors of modern literary science. It is therefore interesting to see how authors of poetics introduce and discuss, e.g., certain recurring topics, styles, or themes that are used in literary works. While this may already be indicated by the logical structure of poetics, e.g., by the chapters and through textual explanations such as chapter titles, it can be missing at finer levels of granularity. Exploration and visual summarization of finer levels can therefore help to understand the content structure within a sub chapter. A more specific question is which authors and which of their writings are referenced as this can explain literary styles and which works are seen as most representative. To achieve the latter goal, an investigation of quotations in poetics seems to be a promising approach and is outlined in the hypothetical example presented in the following section.

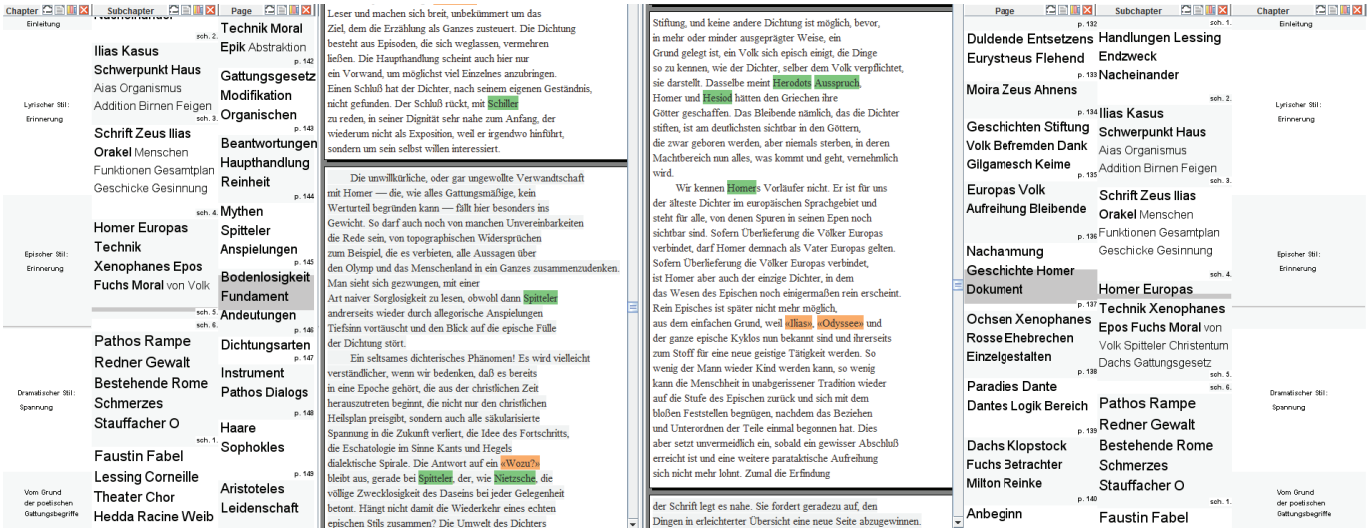

Fig. 7. Two documents can be compared with VarifocalReader by showing them next to each other. Both documents can be navigated independently.

4.1 Finding representative authors for literary genres

To gain some insights into these questions, a fictitious literary scholar might load a digital version of Staiger's "Grundbegriffe der Poetik" into VarifocalReader. The scholar knows that Staiger mainly discusses the three literary genres poetry, drama, and epics in his text book and she is curious which literary works and authors he discusses in the context of dramatic literature. As a first step, she adjusts the layers to show word cloud summarizations in their respective segments. The chapter dealing with dramatic literary works is quickly identified. In the word cloud of this chapter, she notices the title of the literary work "Wallenstein".

She wants to know in which context Staiger discusses this work. Browsing the word cloud summarization of the corresponding subchapters with VarifocalReader indicates that subchapter three is the most promising one to look at. She sees this because "Wallenstein" is among the words with the highest rank in the cloud. After activating the named entity recognition, it is not difficult to realize that several persons are mentioned in this subchapter. Browsing through the names shows that Schiller, the author of this work, and Max Piccolomini, one of the central characters in the second part of Wallenstein, are often and prominently mentioned.

In order to see which works are referenced in this subchapter, the scholar activates the rule-based method for extracting text parts in quotes. She observes that the four literary works "König Ödipus" (Oedipus the King), "Antigone", "Der zerbrochne Krug" (The Broken Jug) and "Prinz Friedrich von Homburg" (The Prince of Homburg) are discussed before Staiger turns to "Wallenstein". She notices that those four works are dealt with in a relatively short passage compared to the following discussion of Wallenstein. This highlights the importance of Wallenstein in this subchapter. By exploring the following text, she learns that Staiger discusses the three parts of the trilogy: "Wallensteins Lager" (Wallenstein's Camp), "Die Piccolomini" (The Piccolomini), and "Wallensteins Tod" (Wallenstein's Death) successively.

Our literary scholar then activates the classification method for distinguishing between quotations, emphases and titles of literary works, which Staiger all writes with double quotes. Inspecting the occurrences identified as quotations, she notices that one is an emphasis rather than a title, changes the class of the annotation, and retrains the classifier. Once the classifier can reliably identify true quotes (as opposed to titles and quotation marks used for emphasis), she can use the bar charts and pictograms on the less detailed layers to be guided to the passages where Staiger most extensively quotes from other works.

4.2 Exploring the Iliad

To show the suitability for English text documents, we present an example based on Homer's "Iliad" using VarifocalReader. The "Iliad" is a classic work about the Trojan war and consequently contains many descriptions of numerous events of this war. An interesting aspect of the "Iliad" is that each chapter begins with a summary of the incidents detailed in the chapter. Accordingly, it would be useful to be able to link the events mentioned in the summary with the text in the chapter describing a particular event.

In our example, a user remembers that there was a fight between Hector, a Trojan prince, and Ajax, one of the heroes of Greece. The user does not remember the details of the fight or its circumstances, therefore browses the chapter headlines, and finds out that chapter seven describes the fight between Ajax and Hector. She is interested in details of the fight itself and begins her investigation by reading the chapter's summary, which brings up two specific questions: i) Did any outside forces interfere during the fight? ii) Who was responsible for Paris's peace-offering being rejected?

To answer those questions, she activates the topic segmentation to get a better overview of the topics discussed in the chapter because the "Iliad" contains no subchapters. In addition to that she activates the named entity recognition. She also activates word clouds on the chapter segments level and displays in parallel the pictogram view and the bar charts on the page level. It becomes obvious that the information she is interested in is most likely contained in the third chapter segment. This is the only segment where Ajax and Hector both appear in the corresponding pictogram. Browsing the occurrences of persons she finds the text passage where the fight is described. She quickly notices that Apollo intervenes in the fight by helping Hector through restoring his strength when he falls to the ground (see Figure 8).

For answering the second question, she recalls the following sentence she has just read in the chapter summary: "Priam sends a herald to make this offer, and to demand a truce for burning the dead, the last of which only is agreed to by Agamemnon." She then notices that the words "truce", "treasure", and "bodies" appear in the world cloud of the third chapter segment. She marks them in the word cloud, looks at their occurrences in the text, and thus finds the text passage which describes how Tydides, one of the Greek, speaks out against the offer of Paris and thus convinces the Greek to reject the peace-offering.

While the usage scenarios show rather specific application examples for VarifocalReader, we think that our approach can be useful for analyzing other literature and long continuous text documents as well. This will be even more the case if additional user-adaptable techniques complement the presented ones. However, even with its

Fig. 8. VarifocalReader showing Homer's Iliad as described in the second usage example. The generated segments are depicted on the left. In the middle, bar charts and the pictogram view are shown. On the right, the plain text with highlighted named entities is depicted.

currently available set of automatic and visual methods, it can address a wide variety of tasks.

5 Discussion

VarifocalReader was developed to support intra-document analysis and not for working with multiple documents at once. However, we see two possibilities to extend the approach in this direction. The first one would be the introduction of an additional parent layer that abstracts multiple text documents or books. This approach, however, introduces an artificial ordering of the documents in the additional library layer. While such an ordering could be meaningful, e.g., by using the documents' creation date, it would necessarily be different from the organization in lower levels of the hierarchy, which are based only on the sequential order of the text. It would therefore break with the concept of depicting the same sequence in each layer.

In its current version, all layers of VarifocalReader are always synchronized. This makes our approach slightly less powerful than other approaches since it is not possible to move to another point of interest in a layer without losing the current text details. However, this was a necessary design choice for the benefit of having many layers. Navigating these could not be handled by users without synchronization. For the class of text analysis tasks we are aiming at, this is no disadvantage since we expect that they always involve reading the text source itself, which requires switching the text details to the new location anyway. Comparing multiple text sources, however, is not well supported by our approach.

In this regard, other approaches that simply show separate text windows are more scalable because multiple text views can be placed next to each other as long as the available display size and resolution allow it. Our approach clearly uses more screen space and, while it supports the comparison of two text documents by showing them next each other with reversed hierarchal foci, it would be difficult to display more than two of them with many layers. Scaling up to multiple documents by introducing an additional library layer as discussed above would not help in such a case. However, by reducing the number and size of the displayed abstraction layers drastically, more documents could be shown next to each other (in this case without mirroring the layers). Depicting just one small additional abstraction would reduce the approach to showing multiple document views with vertical scrollbars holding additional cues. VarifocalReader has its strengths in intra-document analysis. Comparing two text documents works well, but its scalability to more documents is limited.

Scalability is also an issue when working with single documents. Theoretically, documents can be arbitrarily long and screen space is confined. Our approach is scalable because it already abstracts the data at different levels and shows smaller sections of data on more detailed levels. Nevertheless, visual abstractions might be too large for the screen space the user is willing to allocate for them. In such situations, it is still possible to increase the visual aggregation of the shown properties and, e.g., switch from a view that depicts text information using symbolic rows to bar charts showing merely the frequency of a text property.

Another scalability issue arises from the available layers. Our approach relies on available hierarchical structure to present an initial overview of the document. Because these hierarchical levels are either derived from content structure or, if available, layout structure, they vary from document to document. Most often, at least two hierarchical levels are available: the text level is always present and for the medium to large document sizes we are aiming at, pages, sections, chapters, or paragraphs are typically available. There are instances, however, where the intrinsic document hierarchy is skewed in terms of exhibiting strongly varying granularities. This can lead to situations in which the partitioning of hierarchical levels is either too coarse or too fine or the transition between such levels is too drastic.

In order to cope with skewed hierarchies, we included additional mechanisms for splitting large text blocks and adding new intermediate hierarchy levels through topic segmentation as described in Section 3.2.

Our approach particularly supports a hierarchical perspective on sequential text documents. This sequence is maintained automatically between segments within each hierarchical level. However, not all of our visual methods for aggregating text characteristics or summarizing document content reflect the sequential nature of text within the scope of their segment. The line-based view acknowledges the intra-segment sequence of depicted occurrences of text characteristics while others, such as the bar chart, do not. Particularly in combination with the focus highlight (see Section 3.1), this could lead to misinterpretation. Interestingly, this was not a problem reported by our expert user, who also denied to be confused by those visual methods having no sequential alignment within segments. Nevertheless, further investigation might be advisable and long term usage of VarifocalReader by our partners from literary sciences will most probably lead to an authoritative statement regarding this issue.

We discussed advantages and problems of VarifocalReader with an expert from the department of German literature studies at the University of Stuttgart, who worked with our technique. We asked the expert to speak frankly about the approach's benefits and flaws, but it should be mentioned that he is involved in the ePoetics project himself. Subsequently, we only summarize the most interesting feedback. The literature scholar saw the organization of VarifocalReader into different layers as an advantage. He specifically emphasized that the possibility to combine close and distant reading modes supports his tasks and work flows on text documents very well. He considered this to be a unique feature of the approach. When asked to prioritize the usefulness of the available visual representations, he said this would depend on the task and that he uses all variants. Our expert emphasized the benefits of the word cloud representation to be particularly useful for getting a quick overview on a text document. He also mentioned that the word cloud summarizations would help him get new ideas and develop new hypotheses but that they are not useful on the lower layers close to the text layer. He was enthusiastic about the word cloud for the third chapter of Staiger's work, in which he had special interest in. According to him, the word cloud represented the most important concepts discussed in this section of the work very well, and it was straight-forward for him to recapitulate its contents. At this point in time it was possible to create word clouds for each layer independent of the number of contained lines or sentences. We now introduced a threshold of a minimum of 10 per segment for the word cloud option. Accordingly, he remarked that he often uses a setup that shows word clouds on the upper layers and line-based abstractions and bar charts on all layers.

Our expert acknowledged the (semi-)automatic analysis features of the approach, e.g., for adapting uncertain extraction methods. The first time we talked to him, he missed the possibility of comparing texts, which led to the introduction of the mode for comparing two documents. He acknowledged the flexibility to choose which views should be depicted in a layer. Additionally, he liked the option to display the digitized image of a book's page and mentioned that this would increase his trust in the approach. This last aspect was also highly ap-

preciated by the participants of a literature lecture, where the approach was presented. Because of this high interest in the digitized resources, we see the optional augmentation of these images with highlights as a potential direction of future work.

6 Conclusion and Future Work

With VarifocalReader, we present a method that supports visual analytics tasks on large text documents. It is particularly useful in situations where scrutiny is required and findings have to be verified based on the text source. The text part of interest can always remain in focus while, at the same time, our approach lets users exploit document-inherent hierarchical structures to inspect visual summarization on finely graduated layers. This includes the inspection of images of the original pages if the text has been digitized from a physical document. In situations where the granularity of layers is inadequate, additional layers can be generated through topic segmentation and available layers can be either removed or aggregated. Word cloud based summarization enables analysts to understand the most important concepts discussed in corresponding text segments. This is complemented by visual support for conveying the occurrences of findings or extracted passages in the text.

VarifocalReader also offers basic visual and interactive methods for letting users understand and refine machine learning approaches, which are common in natural language processing. The presented approach can be flexibly extended with additional text mining and NLP methods as well as corresponding visual, interactive feedback loops. This is, we believe, most important when the efficiency of high-quality analyses of non-standard text resources is to be increased.

While the approach is already being used in its presented form by literary scholars, there are many possible directions for improvements. Up to now, we have not identified a good presetting or automation for creating and showing an adequate initial set of layers and corresponding visual abstractions within these levels. One reason for this lies in the differences of layout and logical structure of the electronic text documents we are working with in ePoetics and other projects. We hope that long-term usage of VarifocalReader by our partners from the literary studies will create additional insights to find a suitable presetting for a reasonable automatic layer resolution.

Another missing feature is a language guesser. Particularly in poetics, which is our current subject of analysis, authors often cite literary works in their original language. This can conflict with the (automatic) techniques we employ and lead to unintended results. With automatic language detection this can be avoided easily.

In general, we aim at supporting a broader spectrum of NLP techniques as part of VarifocalReader in the future. Many of these techniques rely on part-of-speech (POS) tagging. There are already robust POS taggers available [3, 42], which are based on machine learning techniques and have typically been trained on large, contemporary corpora. However, the performance of these out-of-the-box taggers deteriorates when applied to older or even historic texts such as poetics. Because the number of text resources from earlier time periods that are electronically available is often sparse, unsupervised machine learning techniques cannot be used easily to ensure adequate POS tagging. With VarifocalReader, we see the possibility to follow a different direction by letting users adapt standard POS taggers through interactive feedback and refinement, thereby enhancing the performance of dependent tools.

The visualization techniques we offer to abstract extracted text properties in the different hierarchical layers are up to now suitable to either show occurrences of findings in a text or aggregations of statistical aspects through bar charts or word clouds. How to depict relational or hierarchical aspects in VarifocalReader is a research problem we will address in the future as well.

7 Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) as part of the 'ePoetics' project and the German Science Foundation (DFG) as part of the priority program (SPP) 1335 'Scalable Visual Analytics'.

References

[1] A. Abdul-Rahman, J. Lein, K. Coles, E. Maguire, M. Meyer, M. Wynne, C. R. Johnson, A. Trefethen, and M. Chen. Rule-based visual mappings – with a case study on poetry visualization. *Computer Graphics Forum*, 32(3pt4):381–390, 2013.

[2] Apache Foundation. Apache lucene. `http://lucene.apache.org`, 2014. version 3.5.

[3] Apache Foundation. Apache opennlp. `http://opennlp.apache.org`, 2014. version 1.5.3.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[5] E. Brown, J. Liu, C. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST), 2012*, pages 83–92, Oct 2012.

[6] F. Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 26–33, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[7] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys*, 41(1):1–31, Jan. 2009.

[8] C. Collins, S. Carpendale, and G. Penn. Docuburst: Visualizing document content using language structure. *Computer Graphics Forum*, 28(3):1039–1046, 2009.

[9] M. Correll, M. Witmore, and M. Gleicher. Exploring collections of tagged text for literary scholarship. *Computer Graphics Forum*, 30(3):731–740, 2011.

[10] A. Don, E. Zheleva, M. Gregory, S. Tarkan, L. Auvil, T. Clement, B. Shneiderman, and C. Plaisant. Discovering interesting usage patterns in text collections: Integrating text mining with visualization. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 213–222, New York, NY, USA, 2007. ACM.

[11] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST), 2012*, pages 93–102, Oct 2012.

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

[13] S. Eick, J. Steffen, and J. Sumner, E.E. Seesoft-a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering*, 18(11):957–968, Nov 1992.

[14] A. Endert, P. Fiaux, and C. North. Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2879–2888, 2012.

[15] G. W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '86, pages 16–23, New York, NY, USA, 1986. ACM.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, Nov. 2009.

[17] M. A. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 59–66, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[18] F. Heimerl, S. Koch, H. Bosch, and T. Ertl. Visual Classifier Training for Text Document Retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2839–2848, dec. 2012.

[19] X. Hu, L. Bradel, D. Maiti, L. House, and C. North. Semantics of directly manipulating spatializations. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2052–2059, 2013.

[20] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, editors. *Mastering the Information Age: Solving Problems with Visual Analytics*. Eurographics Association, 2010.

[21] D. Keim and D. Oelke. Literature fingerprinting: A new method for visual literary analysis. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, 2007. VAST 2007.*, pages 115–122, Oct 2007.

[22] J. B. Kruskal and J. M. Landwehr. Icicle plots: Better displays for hier-

archical clustering. *The American Statistician*, 37(2):162–168, 1983.

[23] S. Liu, M. X. Zhou, S. Pan, W. Qian, W. Cai, and X. Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 543–552, New York, NY, USA, 2009. ACM.

[24] F. Moretti. *Graphs, maps, trees: abstract models for a literary history*. Verso, 2005.

[25] D. Oelke, M. Hao, C. Rohrdantz, D. Keim, U. Dayal, L. Haug, and H. Janetzko. Visual opinion analysis of customer feedback data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, 2009. VAST 2009.*, pages 187–194, Oct 2009.

[26] D. Oelke, D. Kokkinakis, and D. A. Keim. Fingerprint matrices: Uncovering the dynamics of social networks in prose literature. *Computer Graphics Forum*, 32(3pt4):371–380, 2013.

[27] D. Oelke, D. Kokkinakis, and M. Malm. Advanced visual analytics methods for literature analysis. In *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, LaTeCH '12, pages 35–44, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[28] D. Oelke, D. Spretke, A. Stoffel, and D. Keim. Visual readability analysis: How to make your writings easier to read. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):662–674, May 2012.

[29] P. Pirolli and S. Card. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. *The Analyst*, pages 2–4, 2005.

[30] S. Ramsay. *Reading machines: Toward an algorithmic criticism*. University of Illinois Press, 2011.

[31] 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*, CHI '94, pages 318–322, New York, NY, USA, 1994. ACM.

[32] B. Renoust, G. Melancon, and M.-L. Viaud. Measuring group cohesion in document collections. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 373–380. IEEE, 2013.

[33] S. Richter. *A History of Poetics: German Scholarly Aesthetics and Poetics in International Context, 1770-1960*. De Gruyter, 2010.

[34] G. G. Robertson and J. D. Mackinlay. The document lens. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*, UIST '93, pages 101–108, New York, NY, USA, 1993. ACM.

[35] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages, 1996.*, pages 336–343, Sep 1996.

[36] E. Staiger. *Emil Staiger: Grundbegriffe der Poetik*. Atlantis Verlag Zürich, 1946.

[37] Stanford CoreNLP. A suite of core nlp tools. `http://nlp.stanford.edu/software/corenlp.shtml`, 2014. version 3.3.1.

[38] J. Stasko, C. Görg, and Z. Liu. Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008.

[39] T. Tenev and R. Rao. Managing multiple focal levels in table lens. In *Proceedings of the IEEE Symposium on Information Visualization, 1997.*, pages 59–63, Oct 1997.

[40] J. J. Thomas and K. A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005.

[41] J. J. Thomas and J. Kielman. Challenges for visual analytics. *Information Visualization*, 8(4):309–314, 2009.

[42] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[43] F. van Ham, M. Wattenberg, and F. Viegas. Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1169–1176, Nov 2009.

[44] F. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, Nov 2007.

[45] R. Vuillemot, T. Clement, C. Plaisant, and A. Kumar. What's being said near "martha"? exploring name entities in literary text collections. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, 2009. VAST 2009.*, pages 107–114, Oct 2009.

[46] M. Wattenberg and F. Viegas. The word tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, Nov 2008.

[47] J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceedings of the IEEE Symposium on Information Visualization, 1995.*, pages 51–58, Oct 1995.

[48] M. Wörner and T. Ertl. Smoothscroll: A multi-scale, multi-layer slider. *Computer Vision, Imaging and Computer Graphics - Theory and Applications*, 274:142–154, 2013.