Martin Baumann, Steffen Koch, Markus John, and Thomas Ertl

Interactive Visualization for Reflected Text Analytics

Abstract: In this chapter, we discuss the contribution of visualization research to the CRETA project. We give a definition of visualization as we conduct it as partners within the context of the project. Then, we introduce the opportunities and challenges of collaboration in interdisciplinary projects involving visualization research and delineate a number of conflicting requirements that need to be balanced in such a setting. The workflow pipeline that is defined by the concept of reflected text analytics can be separated in two phases, roughly subsumed under the notions ‘annotation’ and ‘interpretation’. For each of these phases, we present one approach that supports the respective analysis tasks by means of interactive visualization: an approach dealing with text annotation data by multiple annotators and an approach dealing with networks of characters in narrative texts. We describe the representation and interaction features of these approaches and exemplify their workings in a series of usage scenarios. Finally, we assess our experiences in developing these approaches with respect to the balancing of requirements mentioned above.

Zusammenfassung: In diesem Kapitel erörtern wir den Beitrag der Visualisierungsforschung zum CRETA-Projekt. Wir definieren den Begriff von Visualisierung, so wie wir sie als Partner im Projektkontext betreiben. Dann besprechen wir die Chancen und Herausforderungen bei der Zusammenarbeit in interdisziplinären Projekten, die Visualisierungsforschung einschließen, und wir umreißen eine Reihe von widersprüchlichen Anforderungen, die in einem solchen Umfeld austariert werden müssen. Der Arbeitsablauf, der durch das Konzept der reflektierten Textanalyse definiert wird, kann in zwei Phasen unterteilt werden, die sich in etwa durch die Begriffe ‘Annotation’ und ‘Interpretation’ fassen lassen. Für jede dieser Phasen stellen wir jeweils einen Ansatz vor, der die jeweiligen Analyseaufgaben durch interaktive Visualisierung unterstützt: einen Ansatz, der sich mit Textannotationen von mehreren Annotatoren befaßt und einen Ansatz, der sich mit Figurenkonstellationen in narrativen Texten befaßt. Wir beschreiben die Repräsentations- und Interaktionsmerkmale dieser Ansätze und exemplifizieren ihre Mechanismen anhand einer Reihe von Nutzungsszenarien. Schließlich

Martin Baumann, Steffen Koch, Markus John, Thomas Ertl, Institute for Visualization and Interactive Systems, University of Stuttgart

bewerten wir unsere Erfahrungen beim Entwickeln dieser Ansätze hinsichtlich des oben erwähnten Austarierens von Anforderungen.

1 Ways of Collaboration

In this section, we expound our view on the opportunities and challenges that arise in collaborative research projects between visualization scientists and scholars from text related disciplines (which, in the following are referred to as ‘domains’). Which kinds of research tasks on which kinds of data in these domains could be meaningfully supported by visualization approaches? What characterizes a domain problem as a potentially rewarding subject for visualization research? What do visualization researchers bring to the table, what can be expected from them in terms of expertise and possible solutions – and what not? Figure 1 schematically summarizes the relation between visualization research and domain research in general and within the CRETA project in particular.

1.1 A Definition of Visualization

In order to tackle these questions, we first give a definition of visualization as we conduct it as partners within the context of the CRETA project. In doing so, we build upon the definitions given by Card et al. (1999) and Munzner (2014). We consider visualization as a discipline of computer science, where we conceptualize, implement and evaluate approaches to automatically create interactive, visual representations of abstract data. These representations are designed with the goal to amplify human cognition in order to carry out exploration tasks on large data sets more effectively. In the following, we further unpack and comment some parts of this definition.

By ‘exploration tasks’ we mean research tasks with the goal to detect structures within a data set and formulate hypotheses about their workings. ‘Confirmatory tasks’ that seek to validate or falsify such hypotheses also fall into the reach of our field – even though they are attached less importance – whereas the support of ‘presentation tasks’ that seek to visually communicate facts that are already known about a data set and its structure (e. g. like in the case of newspaper infographics) is not of interest for our research agenda.

The task of free exploration of a mostly unknown, large data set is characteristic for those situations in which visualization can be useful. In the case of a clear-cut research question that can be solved by a purely algorithmic approach, or if the (textual) data set is so small that the respective question can be answered

Fig. 1: Schematic comparison of visualization research and domain research with respect to data and tasks, the developed approaches, and the problems of reflected text analytics

by a thorough close reading, then visualizing the data would create a potentially error-prone overhead. However, for an ill-specified problem, whose tackling requires human world knowledge, intuition, experience, and ingenuity, and that is to be pursued on a data set that is too large in order to handle it without computational support, in such a situation, visualization can provide a means to leverage the human visual apparatus and decision-making capabilities.

In connection with these considerations, the notion of ‘effectiveness’ is also important. In contrast to other instances of crafting visual artifacts, visualization as understood here should produce approaches that aspire not primarily to aesthetic quality, but that put this quality in the service of the higher goods of correctness and accuracy (see Munzner 2009).

The broad notion of visualization as a computer science discipline is commonly further separated into the fields of ‘scientific visualization’, ‘information visualization’, and ‘visual analytics’. Without going into further details, we want to point out a basic differentiating characteristic of these fields. In scientific visualization, data are dealt with that have inherent spatial relations (e. g. the air flow behavior along the surface of a car). In information visualization and visual analytics, however, data are dealt with that are ‘abstract’ in the sense that they lack such inherent spatial relations. Hence, in these fields, a spatial metaphor has to be invented (and learned by the viewer) in order to visually represent the data (e. g. in the case when numerical values are depicted as a bar chart). This abstractness is given in particular for textual data. Hence, when speaking of ‘visualization’ in the following, we refer jointly to the fields of information visualization and visual analytics.

Finally, we would like to emphasize the difference between our visualization research and the research done by domain experts who employ these visualizations to solve a problem in their domain. Thus, while we do research on visualization approaches for these domains, we do not conduct research in the domains themselves. Of course, however, we need to understand the domain tasks in order to conceptualize and implement a visualization solution, and we may need to conduct analysis tasks in order to evaluate our approaches.

1.2 The Interplay of Visualization and Application Domains

Taking up this last point, we now elaborate further on the relation between visualization researchers and domain experts. Munzner (2009) differentiates between, on the one hand, data and tasks that originate in the respective domain and that are described in domain terms and, on the other hand, a set of more generic data and tasks that are described in the vocabulary of the visualization discipline. She

explains that one of the first steps in the creation of a visualization would be to map the raw domain data to their visualization-side abstractions (e. g. by binning or thresholding), and to map the concrete, specific domain tasks to more abstract visualization tasks, which can be organized in a series of general taxonomies. A visualization researcher's goal could then consist in conceiving a visualization approach for dealing with these abstracted data and tasks.

The quest for the generalizability of the developed approach, that is discernible in this process, is characteristic for research efforts in computer science in general and visualization in particular. However, for many domains – especially from the humanities – the data, the analysis tasks, and the approach required to deal with them appear very specific to the respective guiding domain research question at hand. This disparity raises problems for collaborative projects. For the visualization side, it can be unrewarding and expensive to build approaches, that can deal only with one specific set of data and tasks. What visualization researchers typically want is to publish the concept of an approach that is primarily new and secondarily can be proven as useful for the domain problem at hand and potentially for many more related abstract task / data combinations. The domain experts, however, are primarily interested in a useful tool that helps them to solve their problems, and often times, an implementation of a custom-tailored combination of old and time-proven visualization techniques may be the best suited option for their needs.

With domain problems from the humanities, another difficulty can arise from the fact that when problems begin to get interesting for the domain expert, they may be already very hard for the visualization researcher to support, leaving only a small (or sometimes empty) intersection of prospective subjects for collaboration.

Finally, the usage of a visualization prototype for the purpose of design evaluation generally puts lower demands on the implementation quality (in terms of stability, speed, longevity, versatility, accessibility, ease of use etc.) than the usage for the purpose of conducting real, long-term analyses. Furthermore, there may be potential features whose implementation could be very useful for an analyst but that contribute little to nothing to the visualization core idea of the approach at hand. The additional effort required to condition an initial prototype to productive use by a domain expert can be substantial and may provide only little benefit for the visualization researcher in terms of publishable results.

These conflicting requirements of generalizability vs specificity, of different goals with respect to novelty vs usefulness, of what can be done (visualization) and what should be done (domain), and of different expectations with respect to the implementation quality of the visualization approach, can substantially narrow the range of collaboration opportunities. Van Wijk (2006) speaks in this con-

text of an ‘interest gap’ between visualization researchers and domain experts. He goes on to describe a user-centered design approach as the preferable way to deal with this problem. The problems touched in this section are also discussed more recently and specifically from a digital humanities / visualization perspective by Jänicke (2016).

1.3 Visualization within CRETA

In the following sections, we describe two examples of visualization approaches that we designed in the frame of the CRETA project. In doing so, we sought to come up with new and generalizable approaches that are useful for the tasks in the project context as they are laid out in the chapter by Pichler and Reiter (2020), p. 43 ff. in this volume. Furthermore, we sought to implement them in a way that they can be used productively and be further extended in the process of this usage; to this end, both examples are implemented to be used in a web browser.

We support the workflow depicted in Figure 1 on page 44 in the phase of the phenomenon recognition / annotation as well as in the phase of the analysis and interpretation of these data under the guiding questions of the text-related domain disciplines. For the former phase, we developed an approach to support the creators and analysts of annotations in the visual exploration of large annotation data sets (see Section 2). This approach was published in Baumann et al. 2020¹. For the latter phase, we developed an approach for the visual analysis of the temporal development of a text’s network of characters (see Section 3). This approach was published in John et al. 2019.

An essential common feature of both approaches is the close interlocking of the respective visualization (distant views) with the text itself (close views). As mentioned in the chapter by Pichler and Reiter (2020), this allows for the consideration of annotated phenomena within their textual context and for the combined interpretation of quantitative results together with a detailed analysis of particular text passages.

2 Visualization of Annotations

As it was mentioned in the chapter by Pichler and Reiter (2020), the exploratory analysis of text annotations is an important task at several points of the CRETA

¹ Presented at IVAPP 2020, <http://www.ivapp.visigrapp.org/?y=2020>

Fig. 2: The approach with Goethe's *Die Leiden des jungen Werthers*. The overview window (A) with the brush at (D). The annotation window in transit mode with four annotator bands at (B), the annotator IDs at (E), the annotated types' distribution at (F), and a series of name types at (H) (see Section 2.2). The text window (C) with underlined annotations. The filter controls at (G). Interaction: Hovering on the type bar 'Hügel' highlights the respective tokens in the annotation and text windows.

workflow. In the following, we understand a text annotation as a piece of meta information that links a list of tags to a segment of text (i. e. a sequence of tokens). In general, such data can pose severe scaling challenges within multiple dimensions of complexity: The source texts can be long, up to hundreds of thousands of tokens, and bear annotations in similar orders of magnitude. Then – as in the case of the Creta project – there may be multiple annotators, human or algorithmic, who possibly disagree in their assessments. And finally, the annotation themselves can be complex: They may contain multiple tags, or text segments that overlap or whose lengths vary substantially.

Domain tasks that are to be performed on such data can encompass: Analyze the disagreement between the contributing annotators in order to refine and disambiguate the team's annotation guidelines; compare annotations generated by competing algorithms in order to assess systematic differences; or consider high annotator disagreement as an indicator of textual complexity that may warrant further investigations regarding possible patterns in the annotation data and how these patterns may relate to the source text. These tasks can involve a comparison across multiple annotators, the detection of patterns with respect to the annotations' extensions or tags, an analysis of the frequency- and position distribution

of the annotated tokens, or a consideration of annotation segments within their surrounding source text.

Our contribution to deal with such data sets and analysis tasks is an approach to visually and interactively browse text annotations. As can be seen in Figure 2, it consists of three interconnected views: an overview of the whole text (*overview window* at A), a view upon a portion of the annotated text strings (*text window* at C), and an abstract representation of the annotation data in some freely chosen granularity (*annotation window* at B). The latter constitutes the main component, and it shows the data simultaneously in two dimensions: vertically across all contributing annotators and horizontally across the whole token sequence of a freely selected passage of the underlying text. The core concept that we employ in order to deal with the data type's complexity dimensions mentioned above is a combined geometric / semantic zooming mechanism; it governs all other aspects of representation and interaction. We will describe it in Section 2.1 and exemplify its workings in Section 2.2.

We now give a few examples of related approaches that deal with task / data situations similar to ours. For a thorough discussion of the current research embedded into the wider area of text visualization, we refer to Baumann et al. (2020). While these approaches deal with one or more of the complexity dimensions of annotation data that we mentioned in above, none of them encompasses all at the same time. In Table 1, we list them under the aspect as to which of the features they offer that seem important to us in order to support the tasks laid out above.

Correll et al. (2011) present an approach that supports scholars in exploring densely annotated texts. They enhance the scalability by employing a focus+context approach to minimize contexts of less importance with respect to the chosen focus and by allowing to filter dense occurrences of interesting annotations by means of a line chart based abstraction. In the *NEREx* approach, El-Assady et al. (2017) analyze named entities by combining a text with annotation highlights, an abstraction of the text's sentences as lines together with the annotated entities as sequences of glyphs, and a series of node link diagrams that encode relational aspects of the entities. The *VarifocalReader* approach (Koch et al. 2014) is close to ours in that smooth transitional interactions between overview and detail views are supported. However, *VarifocalReader* does not aim at preserving a consistent representation over different levels of abstraction, which we see as an important benefit of our approach for the depicted tasks. Chandrasegaran et al. (2017) present in their visual analytics approach for the support of open coding of natural language texts a series of interconnected views, that closely resemble some of the views we present: a text view with highlights on tokens, a series of overviews showing meta data in the whole text, and an interactive word cloud. They also use similar means of view coordination, allowing the syn-

Tab. 1: Feature comparison of annotation approaches. ○ = unsupported, ◐ = partially supported, ● = supported. Column labels: Range = continuous range of granularities, Annot. = comparison of multiple annotators, Config. = segment overlap or multiple tags, Text = interconnected text view, Token = distribution of annotated tokens, Edit = create and change annotations.

	Range	Annot.	Config.	Text	Token	Edit
Correll et al. 2011	◐	○	○	●	○	○
Widlöcher and Mathet 2012	◐	◐	◐	●	●	●
Landragin et al. 2012	○	○	◐	●	●	●
Koch et al. 2014	◐	○	○	●	●	◐
Meister et al. 2016	○	○	●	●	●	●
Eckart de Castilho et al. 2016	○	◐	◐	●	○	●
Chandrasegaran et al. 2017	◐	○	○	●	●	◐
El-Assady et al. 2017	◐	○	○	●	●	◐
Kleymann et al. 2018	◐	○	○	●	○	○
Baumann et al. 2020	●	●	●	●	●	○

chronization of positional and pattern information across views. However, they neither offer a continuous range of granularities for the annotation abstractions, nor visual support for the display of multiple annotators. The *CATMA* annotation tool uses stacked, colored underlines to encode annotations within the running text (Meister et al. 2016). Its line spacing can be stretched in order to make room for overlapping annotations or annotations with multiple tags whose segments can even be discontinuous. The *Glozz* annotation tool, like most of its competitors, uses highlights on the text to encode annotations (Widlöcher and Mathet 2012). Overlapping segments are shown as overlapping highlights – an approach that does not scale as well as *CATMA*’s stacked underlines. An uncommon feature is the aligner view of *Glozz*. Here, different annotators are arranged in a series of rows, and their annotations are strung as colored bars on a line that represents the text. Quite a unique idea for the encoding of annotations is followed by the design study of Kleymann et al. (2018). Besides using an underlined text, they encode each annotation as a glyph consisting of a vertical bar (representing the text), and a colored horizontal bar (representing the tag, the segment length and the position of the annotation). These glyphs can be arranged freely on a canvas, and upon hovering, a tooltip shows the immediate surrounding text. Further examples of annotation tools employing similar means would be *ANALEC* (Landragin et al. 2012), or *WebAnno* (Eckart de Castilho et al. 2016).

advantages to this way of abstractly representing annotations: First, since there is always a uniform grid of tokens or bins, annotations can be immediately compared across all annotators. Second, since the token strings' lengths are not taken into account, the visual weight of an annotation depends only on the number of tokens that its extension contains. And third, these geometric objects – as opposed to text strings – can be easily stretched or transformed, as we will explain next in connection with our geometric / semantic zooming mechanism.

Fig. 5: Zooming out from left to right. A text passage containing the token ‘dem’ (at the yellow highlight / triangle) in the three zooming modes.

There are three major modes of representation with respect to the number of tokens selected by the brush; henceforth, we call them *atomic*, *transit* and *aggregated* mode (see Figure 5 for a juxtaposition of them). In the atomic and transit modes, the vertical grid objects of the lower and middle layers represent tokens. The main difference between these two modes is that the atomic mode applies when the selection brush is small enough that we can display the text string for each token legibly above the annotator bands (see Figure 5a at A). In transit and aggregated mode however, these token strings are replaced by a depiction of the annotated types, that we explain later. Another commonality of the transit and aggregated modes that distinguishes them from the atomic mode is that the grid units are of uniform width and cannot be generally discerned as discrete objects. When the selection brush gets larger than the tool window’s width contains pixels, the aggregated mode applies, and here the grid units no longer represent tokens but bins of tokens. Within each mode, brushing (or zooming) is a continuous experience since only the selection and size of the shown elements changes (geometric zoom); when crossing a mode border, however, the representation of the annotation information is adjusted in a way such as to optimize the available screen space for the display of the selected data (semantic zoom). In this way, the user

can get an overview of a text of arbitrary length without the need for scrolling (aggregated mode), continuously drill down to a detailed view upon any short selection of text (atomic mode), and always be presented the annotation information of the current selection with the finest resolution at hand.

Fig. 6: Cut-outs of annotator bands in the different zooming modes. In atomic mode, the representation of chunks with less than four annotations is not split.

The problem of displaying constellations where many annotation segments overlap and contain multiple tags is dealt with in different ways in the three zooming modes. In atomic mode and with little overlap, an annotation is encoded as a rectangle spanning its segment's tokens, which is furthermore subdivided into a series of sub-rectangles that are colored according to the annotation's tags (see Figure 5a at C and Figure 6a). For dealing with massive overlap, we introduce the notion of 'chunks', which are chains of annotations by a specific annotator that are linked through their overlap (see Figure 4). When a chunk is too large for the whole stack of its annotation rectangles to be displayed in the available band height in the manner described above, the representation of annotations is split (see Figure 5a at B, Figure 6b): Vertical colored bars on each token encode the number and tags of annotations on this token, whereas thin gray rectangles (or a diagonal line, if a second limit of the stack size is exceeded) indicate the annotations' extension. For the transit mode, we chose to exclusively apply this split representation (see Figure 6c). Finally, in the aggregated mode, the colored bars of the split representation are now used to encode the corresponding information for whole bins, and above them, the number of annotation segments that are cut by a bin is encoded as a number of black dots (see Figure 5c and Figure 6d).

As mentioned above, in the transit and aggregated modes, there is not enough horizontal space to show the text strings of all selected tokens in the upper part of the annotation window. What we do instead is to represent the spatial and

Fig. 7: Types in aggregated mode. Interaction: Hover on ‘Hügel’ next to (A) highlights the type bar and shows 20 local bin positions in the text selection (tick marks at B). Red dots (added for reference) mark names (see Section 2.2).

frequency distribution of the most annotated types within the selection (see Figure 2F and Figure 7). First, in the set of all tokens that are part of some annotation (globally), we create a partition by subsuming all tokens that share a common lemma under this term. For each of these type terms, we then sum up the annotation counts of each of its tokens (the *global type count*), and for each type that contains tokens in the current selection (the *local types*), we sum up the annotation counts of each of its *local tokens*, i. e. the selected tokens, which gives us the *local type count*. In aggregated mode, we furthermore look for those bins, that contain at least one local token (the *local bins*). Next, we sort the local types according to their local and their global type count, select a number of types to be displayed from the top, and compute the ideal middle position of their bars as the median of the middle positions of all of their local tokens or bins. We now go through this sorted selection of types and place each type bar as closely as possible to its ideal position. The bars themselves are comprised of three elements: The gray portion of the bar encodes in its height and in its lightness the local type count. The black frame (which contains a gap for long type strings) encodes in its height the global type count. And finally, the type string, which is shortened to an ellipsis for long strings, is shown in a manner to provide maximal contrast: black on the white background and black or white on the bar, depending on its gray value. Finally, below the type bars at Figure 7B, a series of tick marks indicate upon mouse hover the number and position of a type’s local tokens or bins. This representation of types allows the user to analyze the frequency- and position-distributions of the most annotated tokens in selected passages and compare them to the respective global values.

As mentioned in Section 1.3, an essential part of our approaches is to provide the user with the opportunity to consider the textual data (here: the annotations) within the surrounding text for close reading. In the text window, the text is shown together with annotation marks, and it is connected with the other windows by means of the mechanisms of *scrolling*, *highlighting* and *freezing*. As for scrolling, the window can be scrolled manually, but after altering the brush selection, the most relevant portion of the text is scrolled into view automatically. As for the highlights, they consist of static parts in the text window (colored underlines) and

Fig. 8: Highlighting and freezing interaction. Hover and click the annotation at (C), then hover the token ‘den’ (B).

responsive parts in the text and annotation windows that are interconnected by means of mouse hover actions. The central problem with highlighting in the text is that a token can carry multiple annotations by multiple annotators with multiple tags. Hence, we have to differentiate between hovering over elements that are assigned to a specific annotator and elements that are not. For details on how this is done in the different zooming modes, we refer to Baumann et al. (2020), but in Figure 8 we show a simple example in atomic mode: Here, the static underline encompasses the union of all annotators’ chunks: ‘auf einem Felsen, der an dem Abhänge des Hügels ins Tal’. It is black on all tokens except the first, since the tags are not unique across all annotators. Hovering on the annotation rectangle at (C) darkens it, extends the underline in the text window to a frame, but only on the annotator’s chunk, and highlights the annotation extension (‘dem Abhänge’) with the tag color. Hovering on the token ‘den’ at (B) however, highlights the participating objects of all annotators in the annotation window. Finally, with freezing we mean that the user can lock or unlock the highlights by clicking on an element. Hovering over other elements may then add further highlights, and this state is not lost by changing the selection or by entering into a different zooming mode.

2.2 Usage Scenarios: Annotations of *Werther*

Next, we present two usage scenarios operating on data from the CRETA project. The source text of about 42 000 tokens, the 18th century novel *Die Leiden des jungen Werthers* by Johann Wolfgang von Goethe, was annotated by Sandra Murr (see Ketschik et al. 2020, p. 204 ff. in this volume) and her team of annotators for en-

tities in the categories of ‘person’, ‘location’, ‘organization’, ‘event’, ‘work’, and ‘concept’ (about 6500 annotations).

Fig. 9: The *Werther* data set. Fully zoomed out annotation window in aggregated mode. Interaction: The type ‘Hügel’ at (C) is hovered and the highlights are frozen; at (B), the lens is brought up. Sparse region to the left of (A), dense region at (D).

The first task considered here is to compare the annotations’ tags and extensions across the four annotators and to detect and analyze patterns of annotation. From Figure 9 – showing the whole data set in aggregated mode – we can already deduce some general facts: Mainly ‘person’-entities were annotated (red), the largest type bars correspond to the novel’s main characters ‘Lotte’, ‘Albert’, and ‘Werther’ (see at E), the marks of annotator no. 11 are more sparse (see at A), and there are some quite densely annotated areas (see at D). For the latter, a few quick inspections with the lens (see at B) identify the most interesting candidates for a closer look, and we zoom in at two of them, shown in atomic mode in Figure 5a and Figure 8. Both passages show constellations with a lot of overlap and disagreement. Besides the red ‘person’- and the blue ‘location’-tags, pink tags mark uncertain annotations for resubmission. The tokens carrying the most annotations are ‘St.’ in Figure 5a and ‘ins Tal’ in Figure 8, as can be seen from the dark background color (see at A in both figures respectively). In both passages, annotators no. 12 and 26 (the second and fourth band, counting from top) show a strong agreement, and they were also the ones tagging passages for resubmission. Furthermore, annotator no. 13 (the third band) nested his annotations more strongly in both passages, whereas annotator no. 11 (the first band) annotated more separately. Based on these observations, we can construct a hypothesis describing the annotators’ preferences. We look for further support by first bringing annotators no. 12 and 26 in immediate adjacency and then slide the brush through the overview window at different zoom levels while also keeping an eye

on the brushed passages in the text window. If we can corroborate our conjecture, we may use it to further refine the project's annotation guidelines.

The second task is to analyze the distribution of the annotated tokens and types and their context. Starting again with the overview of Figure 9, we can hover over the largest type bars and see that while for the top three of them – the main characters' names (E) – the annotations are spread all over the text, the annotations of the fourth type – 'Hügel' (C) – are concentrated in a small region towards the end, as the tick marks and yellow highlights indicate. We freeze these highlights by clicking and zoom into a passage large enough to encompass all 'Hügel' occurrences except for a few outliers; this is shown in transit mode in Figure 2. Some of the most annotated types in this passage are character names that appear almost exclusively in annotations in this passage, like 'Salgar', 'Alpin', or 'Ullin' (H). Reading the respective passage in the text window shows that it is part of a short embedded narrative. We enlarge the brush slightly in the overview window, such that it contains the embedded narrative, and can see in the corresponding type bars a series of further character names, that were almost exclusively annotated in this short passage, while the main characters' names do not show up at all (see red dots in Figure 7). Furthermore, 'Hügel' is still the most annotated type here. Based on these findings, it is now an interesting task for an analyst to consider the role that a hill or hills play in this embedded narrative and how the remaining references to hills in the rest of the text relate to that.

2.3 Visual Annotation Analysis in the CRETA Workflow

As it is described in the chapter by Pichler and Reiter (2020) in this volume, the notion of reflected text analytics consists of a succession of interrelated practices that is sketched there in Figure 1 (page 44). An early, large group of these practices is concerned with tagging a set of phenomena on the text surface that operationalize the respective domain problem. With AnnoXplorer, we provide an approach to support the explorative analysis of these annotation data. As shown in the usage scenarios above, this analysis helps to iteratively enhance the model assumptions reflected in the project's annotation guidelines. And it helps to refine the domain research question in the sense of a hermeneutic procedure as described in the introduction of this volume.

3 Visualization of Character Networks

The later stages of the workflow pipeline of reflected text analytics are about interpreting the phenomena that were operationalized in the earlier stages. When the person entities in a narrative text as well as the verbs and adjectives that characterize their relationships are annotated (manually or automatically), one may next want to analyze the ensemble of the plot's characters under the guiding questions of a literary interpretation. Typical domain tasks in this endeavor may encompass to characterize the protagonists of a plot, their attitudes and development, to collect information about the relationships of the different characters and see how they evolve over the course of the plot, and to compare character constellations at different timestamps with each other.

Fig. 10: The main workspace comprises (A) a menu that enables users to switch between the different visualizations or to filter the character network, (B) the main view where users can inspect the character networks with the aid of matrix-based or node-link visualizations, (C) the tab view where users can switch between a graph desktop, an entity list, a text view, and a word cloud view.

Our contribution to support such tasks is an interactive visualization approach that enables users to select different ranges of a text document and to visually analyze and compare the character networks occurring within them. It provides visual abstractions and interactive features like focusing or filtering devices, that facilitate the analysis of information about the change of relationships

between characters as well the analysis of the corresponding text passages. Textual summarizations of the context in which a collection of characters appear characterize their relationships semantically. Here, the concept of ‘relationship’ between two characters and also the concept of ‘characterization’ of a character through a verb or adjective are defined as their respective common mention in a certain text range. The extension of such a range depends on the text considered; in the case of *Parzival* – which we will look at in Section 3.2 – it is one paragraph. The main workspace of the approach can be seen in Figure 10. Its main view in the center displays a matrix-based or a node-link visualization of the character network as appearing in the selected text passages. These text selections are represented as color-encoded brush rectangles in the *document bands* above the visualization. Further auxiliary views are collected in a tabbed panel to the right, and controls for filtering and the mode of representation are collected in a menu to the left. We will describe the details of these representations and interactions in Section 3.1 and again exemplify their usage in a scenario in Section 3.2.

Our visual approach deals with the representation and comparison of graphs as well as with the visual analysis of characters in narrative texts. In the following, we discuss some previous works from these three fields and how they influenced our approach or differ from it. For a more detailed and wider discussion of the relevant research context, we refer to John et al. (2019).

First, let us consider the challenges of the visual comparison of graphs. Gleicher et al. (2011) present a survey that defines three main categories for visual comparison. Based on the defined categories, Beck et al. (2017) slightly revise them for graph comparison challenges. The revised categories comprise ‘juxtaposition’, ‘superimposition’, and ‘integration with explicit visual encoding’. In a juxtaposition approach, the different states of the graph are depicted side by side, whereas a superimposition technique represents the different states on top of each other. In an integrated approach, the different pieces of information about the different states – in our case: mentioned characters in text passages – are an integral part of the visual representation. Some examples for the different techniques are Jianu et al. (2010) for superimposition; Albers et al. (2011) and Blascheck et al. (2017) for juxtaposition; Shi et al. (2011) for integration with explicit encoding; and Federico et al. (2011) for the combination of these techniques.

Second, let us have a look at approaches dealing with the visualization of character networks. These networks differ from social networks – for which there have been derived a series of approaches in recent years – in that they are typically much smaller and show a complex evolution of relationships between their entities. The *NEREx* approach by El-Assady et al. (2017) supports the visual comparative analysis of discussions between several persons by offering a node-link visualization of these persons and their relationships. Kim et al. (2011) present an

approach that combines word clouds and node-link diagrams to visualize named entities and their relationships in text collections. Kurzhals et al. (2016) propose a visual movie analytics approach, which allows users to annotate and analyze characters from movies on different levels of detail. Finally, the *Jigsaw* approach by Stasko et al. (2008) offers multiple coordinated views that enable users to track characters and their relations in large text corpora.

Third, we focus on approaches that integrate similar visualization techniques for visual encodings as our approach. Henry et al. (2007) present *NodeTrix*, which allows users to select different parts of a graph and to seamlessly switch between a matrix-based and a node-link visualization. Cao et al. (2015) present the *g-Miner*, which also supports a matrix-based and node-link visualization to represent network data. Both visualization techniques use similar visual encodings to highlight the differences between variant states: for example a pie chart in the node-link visualization, or bar charts in the matrix-based representation to depict the different numbers of occurrences.

3.1 Character Networks: Representations and Interactions

Let us start with a description of the main view at Figure 10B. Here, users can select one or two text ranges (with the latter, we enter *comparison mode*) by brushing in the document bands and choose to present the respective character networks either as a node-link- or as a matrix-based visualization; or they can choose to display a grid view with selected graphs here (see Figure 12). The gray lines in the document bands represent segments of the document structure like parts or chapters; hovering over them shows a segments' name in a tooltip. Below the document bands, a *sentiment band* can be activated. Here, green bars represent positive and red bars represent negative verbs or adjectives that occur in the context of single characters or of pairs (relationships) of characters. In the main view, we offer four different visual encodings for the node-link diagram and three for the matrix-based visualization in the comparison mode. Furthermore, users can apply filter techniques that support the comparative exploration of the character networks.

The node-link representation is a widely used technique to visualize smaller graphs. It has the advantage to be easily understandable and that paths through the network can be immediately recognized. It consists of vertices (characters) and edges (relationships) to represent the graph structure in a force-directed layout. The four encodings result from a free combination of two encodings for the vertices and two for the edges (see Figure 11). Users can use either a bar or a circle to represent characters. The size of the circles and the length of the bars are scaled

Fig. 11: Four different node-link diagram encodings: (A) juxtaposed bar charts with length comparing edges and (B) pie charts with width comparing edges.

Fig. 12: The grid view provides an overview of all selected character networks as juxtaposed small multiples in an adjacency matrix. The frames' colors represent the similarity between the respective character networks.

proportionally to the number of the characters' occurrences. In the node-link comparison mode (i. e. if two text passages are selected), we map the number of the characters' occurrences in the respective passages to either slices in a pie chart or to bars in a bar chart. For the representation of the relationships, users can choose between length comparing edges and width comparing edges. With the former, the numbers of co-occurrences of the connected characters in the two respective passages are mapped to the lengths of the respective edge segments (Figure 11A); with the latter, they are mapped to the widths of the two parallel lines that compose an edge (Figure 11B). A double-click on a vertex or edge opens the text with the corresponding highlights in the text view, the corresponding related verbs and adjectives in the word cloud view, and the corresponding sentiment results in the sentiment band.

As with all node-link diagrams, the legibility of the character network deteriorates quickly as the number of edges and vertices grows. To alleviate the occlusion problems, we provide filter controls. By hovering over a character, all characters and relations not directly connected to it are grayed out as can be seen in Figure 10. Users can also gray out particular characters and their corresponding relations by clicking on them. Additionally, they can set a percentage to filter characters or relationships that only occur in one of the selected text ranges or that frequently occur in both via the sliders in the menu. However, these filter techniques do only support users up to a certain extent, and therefore, we also provide a matrix-based visualization that eliminates visual clutter.

A matrix-based visualization provides a more scalable approach to represent larger networks. The rows and columns of the matrix represent the characters and the cells their relationships. To represent the number of co-occurrences between

Fig. 14: (A) Our focus+context approach in the comparison mode with two different selected parts of the romance *Parzival*. (B) The graph desktop containing four user-selected graphs.

mode is activated, we represent negative terms in red, neutral terms in black, and positive terms in green. Thus, users can freely explore the word clouds of the characters and their relationships within a single view. If there is not enough space to display a word cloud in a cell, we display one of the above-mentioned visual encodings.

The third representation that can be chosen to be displayed in the main view is the grid view. This view provides an overview of all stored character networks as small multiples as depicted in Figure 12. The rows and columns represent the stored character networks and the cells the comparison networks of them. To indicate the similarity of the networks, we integrate a simple measure that compares the number of characters' occurrences and co-occurrences. Afterward, we map the resulting values to values on a color gradient from red to green and show the color mapping on the respective cell frames. Green colors indicate similar and red colors very different networks. Thus, users receive a similarity overview, which can serve as a starting point for a further analysis.

Next, we want to consider the four auxiliary views that are shown in the tabbed panel on the right (see Figure 10C). The *graph desktop* contains small multiples of graphs corresponding to the text ranges selected by the users, either in the form of a matrix or a node-link representation, depending on the selected mode (see Figure 14B). Here, users can also pan, zoom, and rearrange the small

Fig. 15: The entity list enables users to select and deselect characters from the graph, shows the number of characters' occurrences, and fingerprint visualizations that represents the distribution of the entities in the different parts of the text document.

multiples, and with drag and drop, they can focus the small multiple in the main view.

The *entity list* provides an overview of the annotated and extracted characters (see Figure 15). The list is sorted by the number of the entities' occurrences in the text and users can (de)select them from the network dynamically. In addition, we show a pixel-based fingerprint visualization to represent the distribution of the character appearances in the texts. Each cell depicts, for example, a part or chapter of a text document, and the color saturation represents the number of occurrences. The red frame around the cells indicates the currently selected range of the text document.

Another auxiliary view is the *text view*, which is linked with the matrix-based and node-link visualization. This view is equipped with a vertical fingerprint visualization, which highlights the occurrences (or co-occurrences) of selected (pairs of) characters next to the side bar as depicted in Figure 10C. Blue bars represent parts of the text document, and the (co-)occurrences of the characters are depicted in the color of the respective network. Users can navigate to the corresponding text passages by clicking on an occurrence. Additionally, we also highlight the occurrences of the characters in the text. Thus, users can easily find and inspect the occurrences of the characters in the text.

Finally, the *word cloud view* is also linked with the matrix-based and node-link visualization and enables users to investigate the related verbs and adjectives of the selected characters. In the comparison mode, we again use stacked bar charts to represent the occurrences of the terms in the different selected text ranges. If the sentiment mode is activated, we also color the terms according the aforementioned sentiment assignments.

3.2 Usage Scenario: Characters in *Parzival*

In this scenario we consider an inquiry that was conducted by our project partners in CRETA on the Middle High German text *Parzival* by Wolfram von Eschenbach, which they had annotated manually (see the chapter by Ketschik et al. 2020, in this volume). In order to ensure a better understanding of this inquiry, we reproduce it here on the basis of an English translation of the text by J. Weston, where we extracted the respective entities by means of NLP. Therefore, the use case's results differ slightly from the original inquiry's.

As a first step, the project partners load the first nine books of *Parzival* into our system and start their analysis with exploring the characters in the node-link visualization. During this analysis, they can identify all essential characters of the nine books. Afterward, they want to compare the characters' co-occurrences within the first books with their co-occurrences within the subsequent books. For this purpose, the project partners create two character networks, where they select the first three books and the following four with the respective document bands. For the visual analysis, they switch to the matrix-based representation (see Figure 13) and inspect the character co-occurrences with the different visual encodings.

In a further analysis step, the project partners want to investigate the evolution of the characters' networks over the plot of the first books. With this analysis they want to investigate their hypothesis that the character networks are not similar. Therefore, they create a character network for each of the first four books and store them on the graph desktop as depicted in Figure 14B. Afterward, they switch to the node-link visualization again and activate the grid view to get an overview of the character networks as depicted in Figure 12. With the help of the similarity highlighting, they can easily compare the four networks and confirm that the character networks are not similar.

As a following analysis step, they want to compare the character networks in more detail. Therefore, they focus the different character networks in the main view. They find out that the character networks of the first four books are only connected via a few characters, such as *Parzival* or *Arthur*, as depicted in Figure 11.

Subsequently, they want to further investigate the co-occurrences of the characters. Therefore, they switch to the matrix-based representation and activate the fisheye technique as depicted in Figure 14A. While they freely navigate and analyze the word clouds of the character co-occurrences, they come across some verbs and adjectives that they did not expect. Those verbs and adjectives could now serve as the starting point of a deeper analysis in the text view.

3.3 Visual Character Analysis in the CRETA Workflow

As discussed in the chapter by Pichler and Reiter (2020), the later stages of the reflected text analytics workflow pipeline are concerned with the analysis of the information that was collected in the earlier stages, under the guiding questions of the respective domain. In the case that this information concerns the character entities of a narrative text, these questions aim for such aspects as the characters' attitudes and development, their relations among each other, or the evolution of character constellations over the course of the plot. The goal of detecting unexpected phenomena is well supported by our approach, since it allows to interactively explore this graph data set, and the close interleaving with the source text allows to easily inspect those passages that may best support a hypothesis regarding such a phenomenon.

4 Conclusion

As mentioned in Section 1.3, one central goal we pursued while developing the approaches described here, was to balance the conflicting requirements listed in Section 1.2. As for the novelty aspect, since both approaches were published on renowned visualization conferences requiring approval in a peer-reviewing process, their novelty was assessed favorably by the visualization community. As for the usefulness within the project context, we laid out where in the process of reflected text analytics we sought to offer support, and we exemplified a few ways of usage in the scenarios described in Sections 2.2 and 3.2. However, the effort we were able to allocate to the implementation of our approaches beyond their visualization core ideas was limited, which, in turn, limited their usefulness for our project partners. For example, in its present state, AnnoXplorer is not a fully-fledged annotation tool that would allow for the creation of annotations in a complex multi-user setup, but an approach for the visual analysis of such data. Likewise, the capacities of our approach for the comparison of character networks is limited as far as factors are concerned that are not directly related to its central visualization concepts, such as stability, performance or import capabilities. Finally, the approaches' generalizability was a great concern for us, especially as a connecting factor for future developments. For example, AnnoXplorer is agnostic as far as the source (human or algorithmic) or form (the exact meaning of the tags) of the annotations is concerned. Even the source text does not need to be natural language text at all, but just a sequence of units that can be tagged, such as the sequence of base pairs in a gene. For our character network approach, one cen-

tral contribution is that it allows for the comparison of constellations based on two different text ranges; however, the concept of the approach would allow for the simultaneous comparison of more than just two constellations. For a detailed discussion of the current intentions for these and more future developments, we refer to Baumann et al. (2020) and John et al. (2019) respectively. To sum up our experiences, we think that while it can be difficult to match the initial expectations of the visualization and the domain side – say, with respect to the extent and quality of the implemented software prototypes, or with respect to the availability of the domain data – the learning process resulting from an iterative and collaborative development process can be of great benefit for both sides.

Acknowledgment: The authors would like to thank their research assistants Harutyun Minasyan, Jena Satkunarajan, and David Schütz for their work. They also would like to thank their project partners from CRETA for the valuable feedback and the data provided, especially Sandra Murr and Nora Ketschik and their teams of annotators, as well as André Blessing and Nils Reiter.

References

- Albers, Danielle, Colin Dewey, and Michael Gleicher (2011). “Sequence Surveyor: Leveraging Overview for Scalable Genomic Alignment Visualization”. In: *IEEE Trans. Vis. and Comput. Graph.* 17.12, pp. 2392–2401. doi: 10.1109/TVCG.2011.232.
- El-Assady, Mennatallah, Rita Sevastjanova, Bela Gipp, Daniel A. Keim, and Christopher Collins (2017). “NEREx: Named-Entity Relationship Exploration in Multi-Party Conversations”. In: *Comput. Graph. Forum* 36.3, pp. 213–225. doi: 10.1111/cgf.13181.
- Baumann, Martin, Harutyun Minasyan, Steffen Koch, Kuno Kurzhals, and Thomas Ertl (2020). “AnnoXplorer: A Scalable, Integrated Approach for the Visual Analysis of Text Annotations”. In: *Proc. 15th Int. Jt. Conf. Comput. Vis., Imaging and Comput. Graph. Theory and App. - IVAPP*. Valletta, Malta, pp. 62–75. doi: 10.5220/0008965400620075.
- Beck, Fabian, Michael Burch, Stephan Diehl, and Daniel Weiskopf (2017). “A Taxonomy and Survey of Dynamic Graph Visualization”. In: *Comput. Graph. Forum* 36.1, pp. 133–159. doi: 10.1111/cgf.12791.
- Blaschek, Tanja, Markus Schweizer, Fabian Beck, and Thomas Ertl (2017). “Visual Comparison of Eye Movement Patterns”. In: *Comput. Graph. Forum* 36.3, pp. 87–97. doi: 10.1111/cgf.13170.
- Cao, Nan, Yu-Ru Lin, Liangyue Li, and Hanghang Tong (2015). “g-Miner: Interactive Visual Group Mining on Multivariate Graphs”. In: *Proc. 33rd Ann. ACM Conf Hum. Fact. Comput. Syst. (CHI ’15)*. Seoul, Republic of Korea, pp. 279–288. doi: 10.1145/2702123.2702446.
- Card, Stuart K., Jock D. Mackinlay, and Ben Shneiderman (1999). “Information Visualization”. In: *Readings in Information Visualization*. Ed. by Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. Morgan Kaufmann, pp. 1–34.

- Chandrasegaran, Senthil, Sriram Karthik Badam, Lorraine Kisselburgh, Karthik Ramani, and Niklas Elmqvist (2017). “Integrating Visual Analytics Support for Grounded Theory Practice in Qualitative Text Analysis”. In: *Comput. Graph. Forum* 36.3, pp. 201–212. doi: 10.1111/cgf.13180.
- Correll, Michael A., Michael Witmore, and Michael Gleicher (2011). “Exploring Collections of Tagged Text for Literary Scholarship”. In: *Comput. Graph. Forum* 30.3, pp. 731–740. doi: 10.1111/j.1467-8659.2011.01922.x.
- Eckart de Castilho, Richard, Chris Biemann, Anette Frank, Iryna Gurevych, Silvana Hartmann, Eva Mujdricza-Maydt, and Seid Muhie Yimam (2016). “A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures”. In: *Proc. Workshop Lang. Technol. Resour. and Tools for Digit. Humanities (LT4DH)*. Osaka, Japan, pp. 76–84.
- Federico, Paolo, Wolfgang Aigner, Silvia Miksch, Florian Windhager, and Lukas Zenk (2011). “A Visual Analytics Approach to Dynamic Social Networks”. In: *Proc. 11th Int. Conf. Knowl. Manag. and Knowl. Tech. (i-KNOW '11)*. Graz, Austria, pp. 1–8. doi: 10.1145/2024288.2024344.
- Gleicher, Michael, Danielle Albers Szafir, Rick Walker, Ilir Jusufi, Charles D. Hansen, and Jonathan C. Roberts (2011). “Visual comparison for information visualization”. In: *Inf. Vis.* 10.4, pp. 289–309. doi: 10.1177/1473871611416549.
- Henry, Nathalie, Jean-Daniel Fekete, and Michael J. McGuffin (2007). “NodeTrix: A Hybrid Visualization of Social Networks”. In: *IEEE Trans. Vis. and Comput. Graph.* 13.6, pp. 1302–1309. doi: 10.1109/TVCG.2007.70582.
- Jänicke, Stefan (2016). “Valuable Research for Visualization and Digital Humanities: A Balancing Act”. In: *VIS4DH*. Baltimore, MD.
- Jianu, Radu, Kebing Yu, Lulu Cao, Vinh Nguyen, Arthur R. Salomon, and David H. Laidlaw (2010). “Visual integration of quantitative proteomic data, pathways, and protein interactions”. In: *IEEE Trans. Vis. and Comput. Graph.* 16.4, pp. 609–620. doi: 10.1109/TVCG.2009.106.
- John, Markus, Martin Baumann, David Schuetz, Steffen Koch, and Thomas Ertl (2019). “A Visual Approach for the Comparative Analysis of Character Networks in Narrative Texts”. In: *Proc. IEEE Pac. Vis. Symp. (PacificVis)*. Bangkok, Thailand, pp. 247–256. doi: 10.1109/PacificVis.2019.00037.
- Ketschik, Nora, André Blessing, Sandra Murr, Maximilian Overbeck, and Axel Pichler (2020). “Interdisziplinäre Annotation von Entitätenreferenzen”. In: *Reflektierte Algorithmische Textanalyse*. Ed. by Nils Reiter, Axel Pichler, and Jonas Kuhn. Berlin: De Gruyter, S. 204–236.
- Kim, KyungTae, Sungahn Ko, Niklas Elmqvist, and David S. Ebert (2011). “WordBridge: Using Composite Tag Clouds in Node-Link Diagrams for Visualizing Content and Relations in Text Corpora”. In: *Proc. 44th Hawaii Int. Conf. Syst. Sci. (HICSS)*. Kauai, HI, pp. 1–8. doi: 10.1109/HICSS.2011.499.
- Kleymann, Rabea, Jan Christoph Meister, and Jan-Erik Stange (2018). “Perspektiven Kritischer Interfaces für die Digital Humanities im 3DH-Projekt”. In: *Proc. 5th DHD*. Cologne, DE, pp. 279–283.
- Koch, Steffen, Markus John, Michael Wörner, Andreas Müller, and Thomas Ertl (2014). “VarifocalReader: In-Depth Visual Analysis of Large Text Documents”. In: *IEEE Trans. Vis. and Comput. Graph.* 20.12, pp. 1723–1732. doi: 10.1109/TVCG.2014.2346677.
- Kurzhaus, Kuno, Markus John, Florian Heimerl, Paul Kuznecov, and Daniel Weiskopf (2016). “Visual Movie Analytics”. In: *IEEE Trans. Multimedia* 18.11, pp. 2149–2160. doi: 10.1109/TMM.2016.2614184.

- Landragin, Frédéric, Thierry Poibeau, and Bernard Victorri (2012). “ANALEC: A New Tool for the Dynamic Annotation of Textual Data”. In: *Proc. 8th Int. Conf. Lang. Resour. and Eval. (LREC)*. Istanbul, Turkey, pp. 357–362.
- Meister, Jan Christoph, Marco Petris, Evelyn Gius, and Janina Jacke (2016). *CATMA 5.0: Software for Text Annotation and Analysis*. URL: <http://catma.de/> (visited on Mar. 15, 2019).
- Munzner, Tamara (2009). “A Nested Model for Visualization Design and Validation”. In: *IEEE Trans. Vis. and Comput. Graph.* 15.6, pp. 921–928. doi: 10.1109/TVCG.2009.111.
- Munzner, Tamara (2014). *Visualization Analysis & Design*. CRC Press.
- Pichler, Axel and Nils Reiter (2020). “Reflektierte Textanalyse”. In: *Reflektierte Algorithmische Textanalyse*. Ed. by Nils Reiter, Axel Pichler, and Jonas Kuhn. Berlin: De Gruyter, S. 43–59.
- Shi, Lei, Chen Wang, and Zhen Wen (2011). “Dynamic network visualization in 1.5D”. In: *Proc. IEEE Pac. Vis. Symp. (PacificVis)*. Hong Kong, China, pp. 179–186. doi: 10.1109/PACIFICVIS.2011.5742388.
- Stasko, John T., Carsten Görg, and Zhicheng Liu (2008). “Jigsaw: Supporting Investigative Analysis through Interactive Visualization”. In: *Inf. Vis.* 7.2, pp. 118–132. doi: 10.1057/palgrave.ivs.9500180.
- Van Wijk, Jarke J. (2006). “Bridging the Gaps”. In: *IEEE Comput. Graph. and Appl.* 26.6, pp. 6–9. doi: 10.1109/MCG.2006.120.
- Widlöcher, Antoine and Yann Mathet (2012). “The Glozz Platform: A Corpus Annotation and Mining Tool”. In: *Proc. 2012 ACM Symp. Doc. Eng. (DocEng '12)*. Paris, France, pp. 171–180. doi: 10.1145/2361354.2361394.