Visualizing Variation in Classical Text with Force Directed Storylines

Shejuti Silvia, Ronak Etemadpour, June Abbas, Sam Huskey and Chris Weaver

Abstract—The study of literature is changing dramatically by incorporating new opportunities that digital technology presents. Data visualization overturns the dynamic for literary analysis by revealing and displaying connections and patterns between elements in text. Literary scholars compare and analyze textual variations in different versions of a lost original text and work to reconstruct the original text in the form of a *critical edition*. A critical edition notes textual variations in extensive footnotes, collectively called a *critical apparatus*. Information in the apparatus is of great interest to scholars who seek to explore complex relationships between text versions. Motivated by application to classical Latin texts, we adapted the storyline technique to visualize a critical apparatus. The visualization facilitates guided discovery of similarities and dissimilarities between prior text versions, which are difficult to detect and reason about with traditional deep reading and spreadsheet-based methods. Storyline visualizations help users understand and analyze the interactions between entities in a story and explore how entity relationships evolve over time. Typical design considerations in existing storyline techniques include minimizing line crossing and line wiggling, which are computationally intense problems. Generating storyline layouts in real time is a substantial challenge to interactive visualization. Existing storyline techniques support limited user interaction due to the high cost of layout. We contribute an initial force directed layout algorithm that dynamically reflows storyline layouts with best effort response to internal and coordinated interactions. We anticipate that the characteristics of our layout algorithm will allow for graceful response to a wide variety of interaction types, speeds, and patterns. We conducted a user study to evaluate the legibility of our storyline layout after convergence. The evaluation results demonstrate that most users can accurately complete a wide variety of visual metaphor interpretation, reading, and pattern recognition tasks within 20 seconds.

Index Terms—Storyline visualization, force directed layout, dynamic queries, textual criticism, collation, critical edition, Latin texts

1 Introduction

Original Latin texts from the classical (Roman) era are mostly lost. Only replications of the original texts survive, in the form of manuscripts, early printed editions, and modern editions. Existing manuscripts and the printed editions of a text usually vary significantly due to alterations and errors introduced by scribal and printing processes. Classics scholars refer to manuscripts and early printed editions as *witnesses*. Textual differences between witnesses are called *variants*. Editors *collate* text for a critical edition by carefully choosing variants from the existing witnesses. The critical apparatus consists of a collection of entries for each *lemma*: a word or phrase in the critical edition for which the editor chooses to cite variants from witnesses. The apparatus appears as highly abbreviated footnotes on each page, and requires substantial training to read.

This paper describes application of a highly interactive storyline technique to visualize a single critical edition's apparatus. Students, teachers, and novice readers of Latin texts are able to read and comprehend the visualization even without sufficient training to read the apparatus in its canonical form. For scholars, storyline provides an overview of the entire apparatus to help them detect anomalies and observe patterns above the level of individual lemmas. They can use the visualization to view and analyze the apparatus of an entire critical edition at once, as well as examine individual entries like with a printed edition. The storyline layout works especially well for tracing witnesses of the same lineage (their *stemma*) that have many variants in common throughout the text. The structure of storyline convergence and divergence reveals similarities and dissimilarities between the witnesses. Analyzing the apparatus with storyline might also help scholars understand the editor's conjectures in the collation behind an existing edition, which plays an essential role in creating new and improved editions.

State-of-the-art storyline techniques compute optimal layout with minimum line crossing and line wiggling. These techniques support limited user interaction due to high computational cost of layout. Instead of generating an optimal layout, we focus on generating a layout that is highly interactively responsive yet also topologically good enough for users to perform key reading and pattern recognition tasks with reasonable speed and accuracy. We use a customized force-directed layout algorithm to achieve fast convergence to reasonable layouts regardless of earlier layout states or interaction dynamics. We are developing the new storyline technique to be a component in an upcoming desktop app for visualizing critical editions in the Digital Latin Library. One mode of the app will include brushed highlighting and filtering between storyline elements and the elements in a central view of the main text.

Existing storyline techniques employ combinatory optimization methods that are difficult to apply incrementally and hence do not accommodate interaction well. In contrast, our layout algorithm runs smoothly and continuously in response to interaction, and hence also seems faster (while actually being slower) than existing techniques. We argue that these characteristics allow for graceful response to diverse forms of coordinated interaction, such as dynamic filtering. A user evaluation confirmed that our storyline layout is aesthetically pleasing and easy to read for novice readers, who performed several reading and pattern recognition tasks with accuracy and efficiency not significantly affected by layout complexity.

2 Related Work and Background

For centuries, textual critics and editors have been comparing texts and their provenance of copying to reconstruct original, lost texts. Many digital humanities projects are currently invested in developing tools to facilitate this process. Juxta (`http://www.juxtasoftware.org/`) is an online open-source tool widely used by scholars to visualize textual differences in multiple witnesses and collate them to create new critical editions. Variant Graph [14] represents textual variations as separate paths through a directed acyclic graph, with witness labels and variants positioned on the edges. Due to convoluted layout and the difficulty of effective text positioning, variant graphs have generally poor readability and scalability. Dekker [4] proposed CollateX, a

- *Shejuti Silvia is with University of Oklahoma. E-mail: shejuti@ou.edu.*
- *Ronak Etemadpour is with Oklahoma State University. E-mail: etemadp@cs.okstate.edu.*
- *June Abbas is with University of Oklahoma. E-mail: jmabbas@ou.edu.*
- *Sam Huskey is with University of Oklahoma. E-mail: huskey@ou.edu.*
- *Chris Weaver is with University of Oklahoma. E-mail: weaver@cs.ou.edu.*

*Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx/*

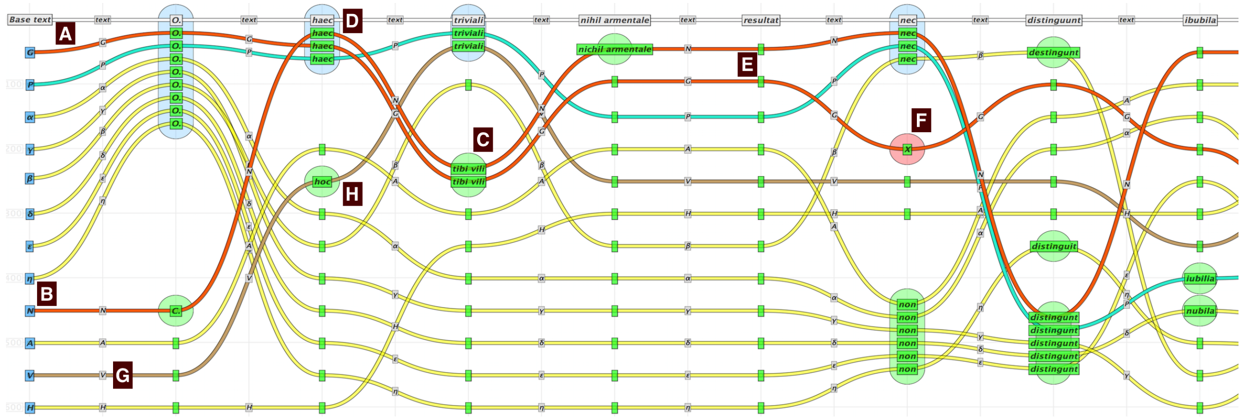

Fig. 1. Storyline visualization of Giarratano's critical edition of the classical Latin poem *Calpurnius Siculus* [17].

modified version of Variant Graph, most commonly used by the scholars for analyzing variants. StemmaWeb [1] is another online application that extends CollateX and provides various methods of analyzing and interpreting textual variants. TRAViz, an interactive implementation of variant graph, aligns sentences from witnesses, based on their similarity in tokens (words or phrases) [8]. TRAViz has proven effective for important application cases like verse-by-verse comparison between Bible editions.

Each line in a storyline represents an entity. Lines flow from left to right, converging or diverging from each other at various points in time, revealing entity relationships. For storyline to be effective, the groupings of lines need to be coherent over time. The ability to show groupings of lines, and how these groupings change over time depending on the evolution of entity relationships, makes storyline a good fit for our application case. Lines represent witnesses, progressing through text (lemmas) in a critical edition, from left to right in reading order. Lines are grouped together when they share the same variant for a lemma. We are particularly interested in tracing patterns of variations between witnesses and tracking how these patterns change throughout the entire text of a critical edition. For our application case, we hypothesize that there is generally enough consistency between groupings of witnesses throughout a text to produce coherent, traceable line grouping. A critical apparatus usually focuses on witnesses from a few reliable lineages, known as *stemma*. Witnesses from the same stemma tend to have many variants in common. A storyline of an apparatus shows patterns that can be used to analyze stemma. For instance, lines converged together into a few groups in a section of a critical edition indicates high quality provenance of the text in that section. Similarly, little coherent grouping with tangled lines between groups indicates substantial *fragmentation* between witnesses. Storyline can also effectively depict anomalies due to *contamination*, in which a line consistently grouped with one group of witnesses throughout the text suddenly diverges and joins another group, revealing its association with multiple lineages.

There is growing interest in automating the layout of storylines [16, 13]. The methods described by Tanahashi, et al. [16, 15] produce storylines for hundreds of entities and event times but take several minutes to lay out, making them too slow for many user interactions including dynamic queries. Storyflow [12] generates layouts faster, enabling it to support fine-tuned interactions such as manual bundling and straightening of lines. Similar time-oriented techniques—ThemeRiver, TextFlow, History flow, TimeNets, popular stack graphs, and layer graphs—effectively reveals different kinds of temporal structure and dynamics [7, 3, 18, 10, 2, 5, 11].

3 Design

Our primary design goal was to provide a visualization of the interaction between witnesses throughout a text. With the storyline representation of the apparatus, we aim to detect certain patterns and anomalies between witnesses providing variants. Figure 1 shows a storyline visualization of the critical apparatus from a classical Latin poem *Calpurnius Siculus* [17]. We implemented our layout algorithm and visualization by adapting the existing general-purpose graph view in Improvise [19, 9]. The transparent line on top represents text in a critical edition (A). Lemmas are equally spaced on the top line, from left to right, in reading order. Each line (except the top line) represents a witness. Each variant of a lemma is vertically aligned with the lemma, and horizontally positioned on the contributing witness line. Multiple lines are grouped together in a "blob" or "pack" when they have a common variant (C). Green, blue, and red blobs represent three variant categories of *same as lemma* (D), *variant* (H), and *omission* (F), respectively. Lines of the same color represent witnesses from the same *stemma*. Each line is labeled with their witness name (B). Empty boxes indicate no textual variation (E).

The challenge to effective application of storyline in our case is that grouping of witnesses specifying common variants (C) does not tend to be coherent across consecutive vertical slots, i.e. lemmas in reading order. In contrast, entities in traditional storylines generally interact with each other over relatively long periods of time, causing the lines to remain together across multiple slots along a time line. The lines in our layout also remain throughout the entire text by design, unlike in many storyline applications, in which entities can appear and disappear over time, or even appear in discontinuous line segments. Overall we are mapping more complex data into a storyline layout than in existing narrative storyline tools.

4 Layout Algorithm

We aim to develop a storyline layout algorithm that dynamically generates decent layouts at interactive speeds, and thereby support common interactions like dynamic filtering. This motivated us to develop a new storyline rendering pipeline built around a force directed layout (FDL) algorithm. We use a modified version of the Fruchterman-Reingold model [6] for our layout. In our force model, a node represents a variant of a lemma at a particular point in the text. Edges are connections between variants for a given witness. Hyperedge 'blobs' are sets of witnesses that have a common variant for a lemma. Application of storylines to critical editions is thus a special case of graph layout. All forces and node movements are vertical only. The variant nodes for each lemma are initially placed in a column, with columns laid out from left to right for each lemma, in text order.

The layout algorithm repeats two major steps: an organic step and a fixed step; see Figure 2. In the organic step, we apply Algorithm 1) for 30 iterations to converge to a minimum-crossing topological layout. The force model used in the organic step is as follows (see Figure 3):

Node-node (n-n) forces: Nodes in a column attract and repulse each other using an inverse-squared force with equilibrium distance. This force groups and separates variants in each lemma's column.

Node-edge (n-e) forces: Nodes connected by an edge (along a story-

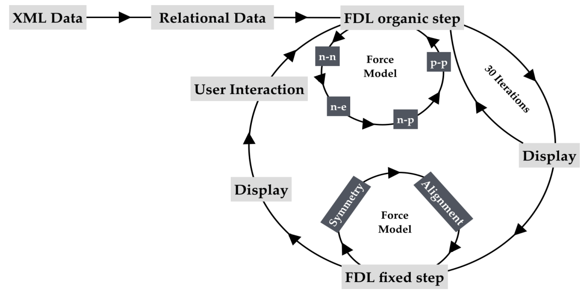

Fig. 2. Data and interaction flow pipeline for our FDL storyline layout.

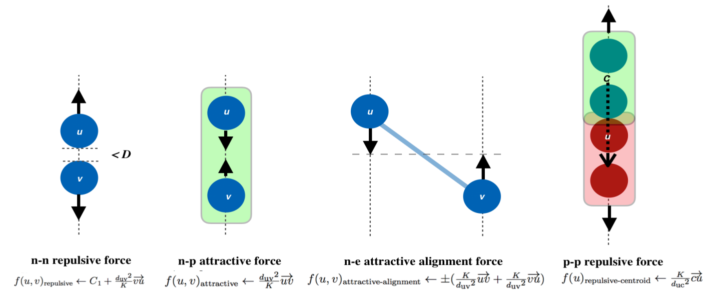

Fig. 3. Force model used in the FDL organic step of our storyline layout

line) attract each other vertically, thus pulling the nodes to align horizontally. This force reduces line crossing and line wiggling.

Node-pack (n-p) attractive forces: Nodes within each variant pack attract each other. This force helps clustering witnesses (lines) providing the same variant, within each lemma's column.

Pack-pack (p-p) repulsive forces: The packs in a column repulse each other, using an inverse-square force on their centroids. This force separates witnesses (lines) providing different variants within each lemma's column.

Algorithm 1 FDL Organic Step

```
for every vertical slot s do
    for every pair of nodes u, v on s do
        if Euclidean dist. between u, v < min dist. threshold D then
            apply n-n repulsive force on u, v
        else if u, v is in a pack p then
            apply n-p attractive force on u, v
        end if
        if a non pack member node u overlaps with a pack p then
            apply p-p repulsive force on u
        end if
    end for
    for every pair of nodes u, v on vertical slots s, s+1 do
        if u, v connected by an edge then
            apply n-e attractive alignment force on u, v
        end if
    end for
end for
```

At the end of 30 iterations of the organic step, we apply strong local symmetry and alignment forces for one more iteration to improve the aesthetics of the storyline layout. We apply the steps and their forces iteratively and continuously. In each iteration, we combine the forces to achieve a good balance in convergence to reasonable layouts regardless of the previous layout state. This approach allows for a graceful response to a wide variety of interaction types, speeds, and patterns. This interactive behavior will in turn greatly facilitate the flexible design of coordinated multiple view visualizations that include the new storyline. The number of iterations in the organic step and all constants used in our force functions were chosen empirically

In generating a storyline graph, each organic step computes in $O(L * W^2)$ time, which dominates the overall time complexity of iteration in the algorithm. It takes under 2 seconds to converge the storyline graph in Figure 1 (not entirely visible), with 494 nodes and 962 edges, using a MacBook Pro with an Intel Core i7 processor. Most of this time is due to rerendering the graph after each iteration, rather than waiting until convergence to render once.

5 Evaluation

We conducted a user study to evaluate the legibility and aesthetics of our storyline technique. We tested the following hypotheses:
H1 *Our storyline visualization is easily readable to novice readers.*
H2 *Our storyline visualization demonstrates clear separation and groupings of entities based on entity relationships.*
H3 *Our storyline algorithm generates topologically good layout.*
H4 *The density and complexity of our layout affects readability.*

5.1 Procedure

Nineteen students from a variety of majors participated in this study. To recruit participants, we sent word-of-mouth invitation and email announcements through a campus mailing list. We conducted a separate study session for each participant. The study lasted for an hour and consisted of two sessions with a five-minute break in between. During the first session, participants performed various reading tasks using the storyline layout. In the second session, participants answered a set of qualitative questions regarding the legibility and aesthetics of the layout. The study began with the participant signing the consent form and completing a background survey. After that, we provided a brief introduction to our visualization and demonstrated various reading tasks using our layout. To minimize the influence of learning, we asked participants to perform several training tasks similar to the ones they would be performing in the study. Once the participant was familiar with our visualization, we provided a questionnaire with 18 quantitative tasks. We used two different storyline layouts in this study. One of the layouts was denser (with 50 variants and 13 lines) than the other (with 25 variants and 7 lines). Each participant was presented with either of the layouts. The study was conducted in an isolated room with the participant seated in front of an Apple MacBook Pro with a 15" screen displaying a storyline layout. Participants used typical interaction techniques like pan and scroll with mouse and keyboard. After completing the first session, participants took a five minute break. During this second session the participant provided feedback to 7 qualitative questions. All questions required choosing an answer from multiple choices or supplying a brief numeric or text response. Given ground truth, we computed the error in a participant's answer to each quantitative question as zero or one. Participants also provided their confidence level in performing each of the tasks on a 5-step Likert scale. We used a stopwatch to measure participant response time and recorded responses using textual transcription and audio recording.

5.2 Task and Data Analysis

Three analyses were of primary interest: first, whether the visual metaphors used in storyline to represent various entities were intuitive; second, whether the various entity relationships were easily readable using the layout; and finally, whether it was possible to find patterns or similarities between entities using the storyline layout. Correspondingly, we divided the quantitative tasks into three categories: *visual metaphor interpretation*, *reading*, and *pattern recognition* tasks.

Visual metaphor interpretation (category 1) tasks required participants to identify lines and nodes representing witnesses and lemmas. Participants demonstrated very high mean accuracy (Figure 4(Left)), confidence, and speed in performing these tasks, with both the dense and sparse layout. Figure 4(Right) shows most participants completed these tasks within 20 seconds with 100% accuracy. We found strong positive correlation between accuracy and confidence, and moderate negative correlation between speed and confidence, for these tasks.

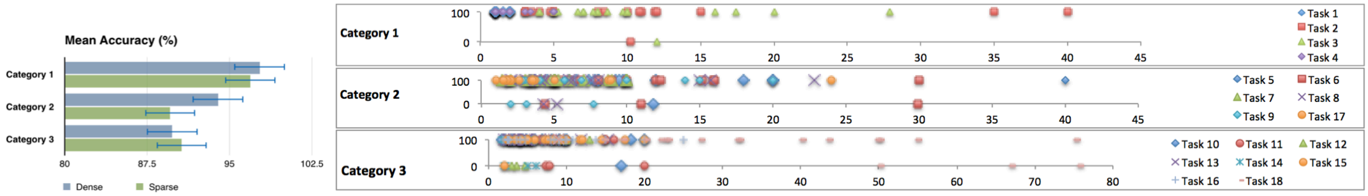

Fig. 4. Left: Mean accuracy for 3 categories of quantitative tasks with the dense and sparse storyline layouts. Right: User accuracy versus time taken to perform visual metaphor interpretation (category 1), reading (category 2), and pattern recognition (category 3) tasks.

Reading (category 2) tasks required participants to identify variants for lemmas, categories of variants, and witnesses contributing those variants. Participants showed moderately high mean accuracy (Figure 4(Left)), confidence, and speed in performing these tasks, with both the dense and sparse layout. Figure 4(Right) shows most participants completed these tasks within 20 seconds with 100% accuracy. A few participants demonstrated poor accuracy and confidence reading variants with similar spelling due to small font size. We found strong negative correlation between speed and confidence for these tasks.

Pattern recognition (category 3) tasks required finding groupings of lines in blobs, indicating common variants between witnesses. Participants demonstrated good mean accuracy (Figure 4(Left)), confidence, and speed performing these tasks with both the dense and sparse layouts. As shown in Figure 4(Right), most of the tasks were completed within 20 seconds. Task 18 required participants to trace a set of lines converging at various points through the layout, indicating their similarity in providing variants. Due to the higher complexity of the task, participants took more time completing this task. We found strong to moderate negative correlation between speed and accuracy, and between speed and confidence, and moderate positive correlation between accuracy and confidence for these tasks.

We performed a Wilcoxon Rank-Sum test on two unpaired samples of performance, on the dense and sparse layouts, for each of the three categories of quantitative tasks. All test results showed p values much larger than 0.05, indicating accuracy, confidence, and speed distribution is identical for both of these layouts.

To analyze the responses to the qualitative questions, we divided the questions into two categories–category 1 questions were related to the aesthetics, spacing, density, and complexity of the storyline layout. All participants agreed that the storyline layout was appealing to them with an average rank of 4.2 (1=least appealing, 5=most appealing). Participants provided feedback such as: they really liked how lines were grouped; the lemmas and the variants were nicely spaced; and overall the layout was clear and easy to read. A few participants who used the dense layout observed that some parts of the layout had more line crossings, which made it difficult to read the line labels. Overall, participants gave an average rating of 3.7 for the spacing, density, and complexity of the layout (1=low, 5=high). Category 2 qualitative questions asked participants to identify the quantitative tasks that were relatively easy or difficult to perform using the storyline layout. Many participants stated that identifying the text line (transparent) and omission for a lemma (red blobs) were the easiest. Many participants found that tracing groups of lines throughout the layout was relatively difficult, especially in places with many line crossings and line wiggles.

5.3 Findings

In this section we relate both quantitative and qualitative findings of the study to our hypotheses and draw conclusions.

Quantitative analysis shows high mean performance for reading and visual metaphor interpretation tasks. Most participants completed these tasks within 20 seconds with very high accuracy and confidence. A few participants had difficulty reading due to the smaller font size displaying the text. None of the participants had prior experience with Latin. Therefore, we accept H1 and conclude that our storyline representation of the critical apparatus is easily readable to novice readers.

The pattern recognition tasks were designed to evaluate H2. Quantitative analysis shows good mean performance for these tasks. A few participants reported line crossing caused difficulty tracing lines throughout the layout. Most of these tasks were completed within 20 seconds with very high accuracy and confidence, which is consistent with all 3 categories of tasks. However, participants took a longer time to accurately perform tasks of higher complexity. Therefore, we claim H2 holds true and conclude that our storyline visualization demonstrates clear separation and groupings of entities based on entity relationships.

Instead of generating an optimal layout with minimal line crossing and line wiggling, which is computationally expensive, our FDL algorithm produces a topologically good enough layout, which allows users to perform various tasks with reasonable speed and accuracy. Our quantitative analysis indicates participants were able to perform reading and pattern recognition tasks with high mean accuracy, confidence, and speed. Our qualitative analysis demonstrates participants found the storyline layout to be aesthetically pleasing and nicely spaced with moderate density and complexity. Therefore, we can accept H3 and conclude that even though our layout does not meet the standard of an optimized layout, it is a sufficient layout for readability.

To evaluate H4, we ran a Wilcoxon Rank-Sum test on performance data drawn from two samples (dense and sparse) for each of the three categories of quantitative tasks. The results indicate no significant difference in accuracy, confidence, or speed for tasks performed with either the dense or sparse layout. Therefore, we reject H4 and claim that density and complexity of the layout does not affect the readability of our storyline visualization.

The results from our user study demonstrates that our storyline visualization was easy to use for novice readers, who performed various reading and pattern recognition tasks with accuracy and efficiency that was not significantly affected by the complexity of the layout.

6 Conclusion

This paper presents outcomes of an initial phase of ongoing research on highly interactive storyline visualization. Before moving to the next phase, we set out to investigate the legibility and effectiveness of the current storyline layout. The next phase of the research includes implementation of user interactions to dynamically generate inputs to the storyline layout in real time, followed by an extensive usability evaluation of how interaction affects the usability of the layout. Representing the text of a critical edition as the top line within the same view as the witness lines causes the text line to disappear during vertical scrolling, affecting scalability. In the future, we plan to use a separate view for the top line to improve the scalability of our storyline layout. We anticipate that the dynamic querying capability of our storyline visualization will help users browse and dissect detailed relational structures in data like that of a critical apparatus. Highly dynamic querying calls for layout techniques that dynamically generate sufficiently good layouts at interactive speeds. Current storyline techniques do not produce layouts quickly enough to accommodate more than limited interaction capabilities. The work described here represents a step toward a general-purpose storyline technique that allows visualization designers and users to interact with appealing layouts in more complex and integrated multiview visualization designs.

Acknowledgments

This work was supported as a part of the Digital Latin Library project by a grant from the Andrew W. Mellon Foundation.

References

[1] T. L. Andrews and C. Macé. Beyond the tree of texts: Building an empirical model of scribal variation through graph analysis of texts and stemmata. In *Literary and Linguistic Computing*, volume 28, pages 504–521, December 2013.

[2] L. Byron and M. Wattenberg. Stacked Graphs - Geometry & Aesthetics. In *IEEE Transactions on Visualization and Computer Graphics*, pages 14(6), 1245–1252, 2008.

[3] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. TextFlow: Towards Better Understanding of Evolving Topics in Text. In *IEEE Transactions on Visualization and Computer Graphics*, pages 17(12):2412–2421, 2011.

[4] R. H. Dekker and G. Middell. Computer-Supported Collation with CollateX: Managing Textual Variance in an Environment with Varying Requirements. In *Supporting Digital Humanities*, pages 17–18. University of Copenhagen, Denmark, 2011.

[5] M. Dork, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. In *IEEE Transactions on Visualization and Computer Graphics*, pages 16(6):1129–1138, 2010.

[6] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. In *Software- Practice and Experience*, volume 21, pages 1129–1164, November 1991.

[7] S. Havre, B.Hetzler, and L. Nowell. ThemeRiver: In Search of Trends, Patterns, and Relationships. In *IEEE Transactions on Visualization and Computer Graphics*, pages 8(1):9–20, 2002.

[8] S. Janicke, A. Geßner, M. Buchler, and G. Scheuermann. Visualizations for Text Re-use. In *International Conference on Information Visualization Theory and Applications (IVAPP)*. 59-70, June 2014.

[9] T. Jankun-Kelly, T. Dwyer, D. Holten, C. Hurter, M. Nöllenburg, C. Weaver, and K. Xu. Scalability Considerations for Multivariate Graph Visualization. In A. Kerren, H. C. Purchase, and M. O. Ward, editors, *Multivariate Network Visualization*, volume 8380 of *Lecture Notes in Computer Science*, chapter Scalability Considerations for Multivariate Graph Visualization, pages 207–235. Springer, 2014.

[10] N. W. Kim, S. K. Card, and J. Heer. Tracing genealogical data with timenets. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 241–248, 2010.

[11] J. Leskovec, L. Backstrom, , and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09*, pages 497–506, New York, NY, USA, 2009. ACM.

[12] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. StoryFlow: Tracking the Evolution of Stories. In *IEEE Transactions on Visualization and Computer Graphics*, volume 19, pages 2436–2445, December 2013.

[13] M. Ogawa and K.-L. Ma. Software evolution storylines. In *Proceedings of the 5th international symposium on Software visualization*, pages 35–42. ACM, 2010.

[14] D. Schmidt and R. Colomb. A data structure for representing multiversion texts online. In *International Journal of Human-Computer Studies*, 67(6), pages 497–514, 2009.

[15] Y. Tanahashi, C.-H. Hsueh, and K.-L. Ma. An Efficient Framework for Generating Storyline Visualizations from Streaming Data. In *IEEE Transactions on Visualization and Computer Graphics*, volume 21, pages 730–742, 2015.

[16] Y. Tanahashi and K.-L. Ma. Design Considerations for Optimizing Storyline Visualizations. In *IEEE Transactions on Visualization and Computer Graphics*, volume 18, pages 2679–2688, December 2012.

[17] Titus Calpurnius Siculus and Caesar Giarratano. *Calpurnii Et Nemesiani Bucolica*. 1910.

[18] F. Viegas, M. Wattenberg, and K. Dave. Studying Cooperation and Conflict between Authors with history flow Visualizations. In *CHI,Vienna, Austria,*, 2004.

[19] C. Weaver. A User Interface for Interactive Construction of Highly-Coordinated Visualizations. In *University of Wisconsin-Madison*, June 2006.