5 Design Rules for Visualizing Text Variant Graphs

Stefan Jänicke[1], Annette Geßner[2], Marco Büchler[2], and Gerik Scheuermann[1]

[1]Image and Signal Processing Group, Institute for Computer Science,
Leipzig University, Germany
[2]Göttingen Centre for Digital Humanities, University of Göttingen, Germany

1 Motivation

After Schmidt's article [1] on modelling and representing various versions of text with so called Variant Graphs was published in 2009, web-based tools were developed that utilize and adopt the presented model to facilitate the work with digital editions of text in the browser. CollateX [2] is one of these tools. It computes a static, horizontally aligned, directed acyclic graph with vertices showing the various text fragments and edges labeled with edition identifiers connecting subsequent text fragments. The tool Stemmaweb [3] extends the CollateX graph to allow for user-driven annotation and modification of the graph structure (e.g., merging and splitting of vertices). Despite the attached interaction capabilities, it seems that there is little value put on designing the graphs. The purpose of this paper is to raise awareness for improving the readability of Text Variant Graphs. We propose a list of design rules for styling the graph, and its vertices and edges to facilitate a rapid comprehension of the underlying alignment structure by the user.

2 Design Rules for Text Variant Graphs

When defining rules for the layout of Text Variant Graphs, we refer to the visualization of CollateX, as its layout is also used in Stemmaweb, and it can be seen as an improvement in comparison to the generated graph of the tool NMerge [4] – provided by Schmidt –, where edges carry all types of information.

When we take a look at a resultant CollateX graph (see Figure 1), it is hard to find out how often a text fragment occurs over all given editions. Thus, it is hard to compare, e.g. the numbers of synonyms. The only chance is to count the edition identifiers at the labels of the incoming edges of the desired vertices. But we can easily put this information on the vertex layouts, which leads us to **Rule 1: Vary vertex label sizes!** As Wattenberg proposes for the "Word Tree" [5], we suggest weighted vertices also for Text Variant Graphs. The usage of font size as a metaphor to reflect the number of occurrences of individual text fragments helps to immediately differentiate between frequent and infrequent branches of the graph.

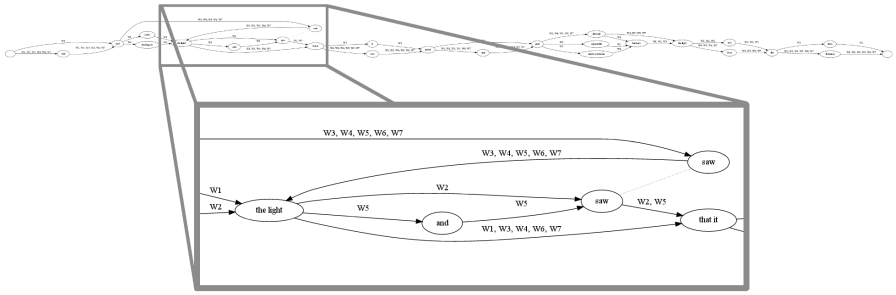

Fig. 1. Fourth Bible verse in 7 different editions: Visualization computed with CollateX tool

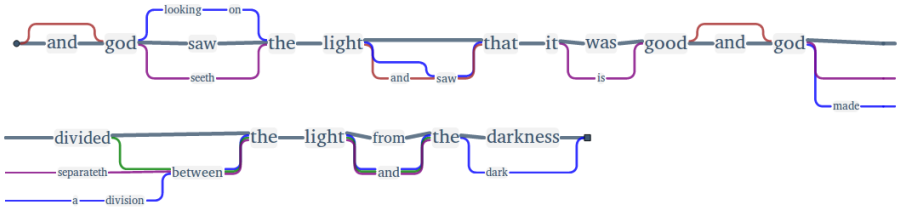

Fig. 2. Fourth Bible verse in 7 different editions: Our visualization

It seems to be obvious to draw the edges of a directed acyclic graph in the shape of arrows. But for what reason, when we know that we read a text with a dedicated writing direction? Then, most probably, the user is supposed to read the graph in the same direction, and it is counterintuitive to move the eyes backwards when reading (in Figure 1, we find an edge from "saw" (right-top) to "the light" (bottom-left) – we call this a backward edge). In graph theory, the common style for a directed acyclic graph is a so called layered graph drawing [6] with all edges pointing in the same direction. Thus, we define **Rule 2: Abolish backward edges!** When doing so, we can reduce the cognitive load of the visualization by drawing undirected edges instead of arrows.

The labeling of edges with edition identifiers as it is done in CollateX leads to two problems. Firstly, the additional text labels interfere with the vertices' texts. Thus, the reader has to visually separate vertex labels (text fragments) from edge labels (identifiers). Secondly, if lots of editions pass an edge or long edition identifiers are used, the corresponding edge labels become very large. As a consequence, adjacent vertices drift apart and the reader quickly loses the context of a text fragment. Therefore, follows our **Rule 3: Do not label edges!** As an alternative, we suggest drawing an edge for each edition in a different color. However, as the human ability to distinguish colors is limited, it only works well for a small number (<10) of editions. But, with varying stroke styles for edges (e.g., line, dashed line, dotted line) we are again able to increase this number. In

any case, a legend is required to map the given styles (or in the CollateX case, the identifiers) to its corresponding edition name.

When analyzing and comparing text editions to each other, the user is often interested in those editions, that deviate from the "general case". Within the Stemmaweb tool [7], edges, that are passed by most editions, are labeled with "majority", thus, the labels are bundled. When following Rule 3, we receive multiple lines instead of multiple labels, hence, **Rule 4: Bundle major edges!** We highlight both resultant edge types – bundled and unbundled edges – in a different way. Unbundled edges receive saturated colors in visually attracting hues, whereas bundled edges are colored in a plain gray, but drawn with slightly thicker strokes. Thus, the deviations from the general case can be detected easily. When following Rules 3 and 4, we are able to reduce the number of edges to be drawn – and therefore, the cognitive load of our approach – to a passable minimum.

Last but not least, the main problem we identified when reading the horizontal aligned Text Variant Graphs was the required horizontal scrolling. Especially, when the source texts are long, the user quickly loses the context and it is hard to keep track of how individual editions disseminate in the graph. Moreover, a lot of space is wasted since the height of the graph is rather small compared to the height of the screen. The outcome of a survey within the TAdER Project [8] to avoid horizontal scrolling when reading texts in the browser underpins our hypothesis that the user is accustomed to scroll vertically. Thus, here comes our final **Rule 5: Insert line breaks!** It may sound tricky to cut the graph into pieces, and thereby, keeping it easily readable. But, why shouldn't we adopt the behavior of a text flowing in a book (with line breaks) for Text Variant Graphs? When following Rule 3, we receive different colored edges (or edge bundles) at the end of each line, so that all paths are visually seizable at the beginning of the next line. This approach supports the user in following individual editions even for large graphs, and the user also receives more context on the screen for a specific position in the graph.

Figure **??** juxtaposes the resultant Text Variant Graphs for the fourth Bible verse with CollateX and our visualization that implements the listed design rules above.

3 Following the Rules: 7 English Translations of the Bible

We are working with seven english translations of the Bible in our project [9], which turned out to be a very good use-case for the presented visualization approach, not only because the Bible is a very influential and well-known text. Another reason is, that these translations are all derivatives of the same Hebrew and Greek original, often trying hard to preserve the exact wording, and refer to an existing and well structured text, divided into canonical books and verses.

Figure 3 shows a screenshot of the first five Bible verses. After tokenization, normalization and alignment procedures, we layout the resultant directed acyclic graph by following the rules proposed in the previous section. For the seven

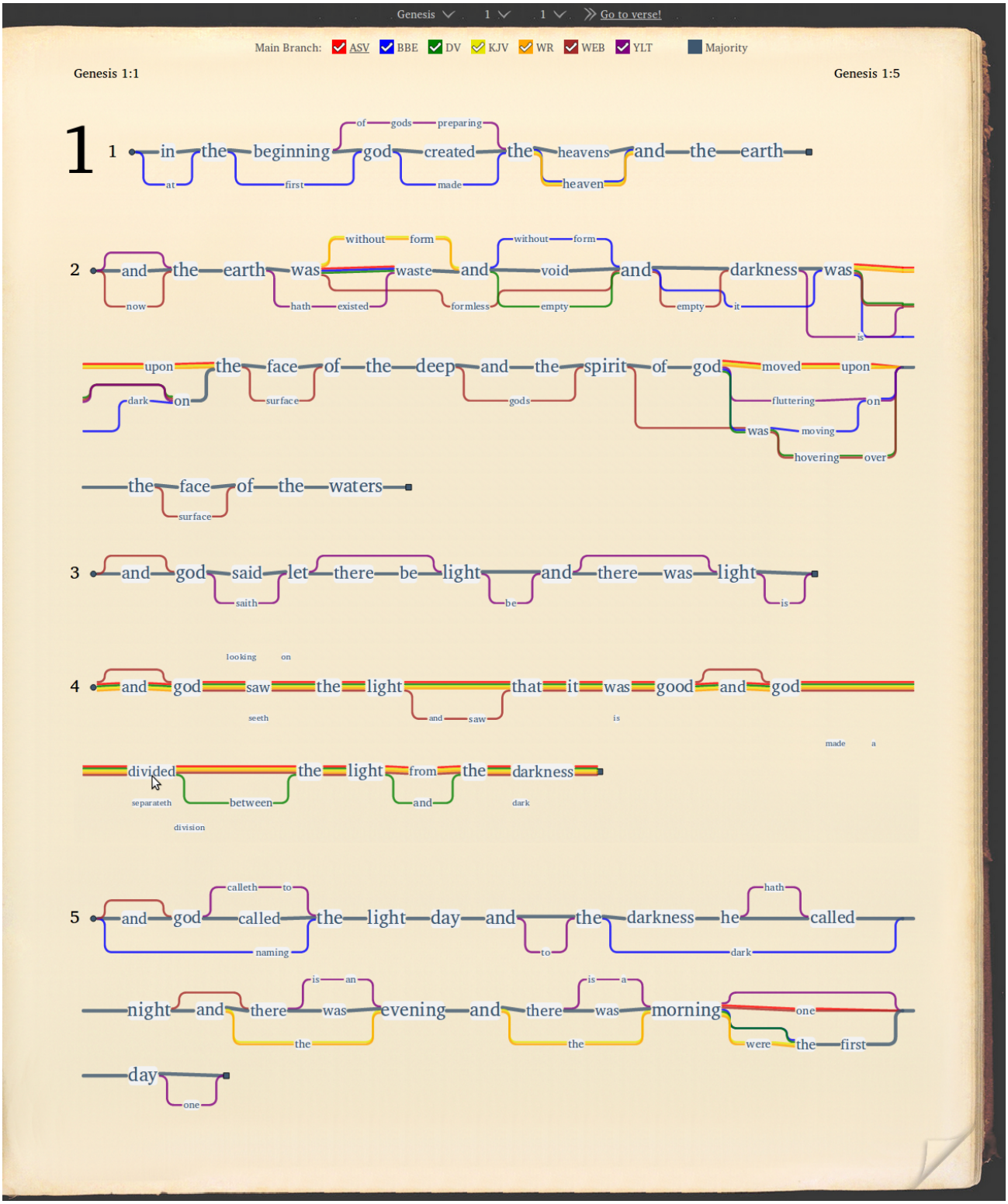

Fig. 3. Screenshot of the Bible use case. In *Genesis 1:4*, all paths containing the token "divided" are highlighted.

editions, we chose the following colors of the 12-color palette for categorial usage suggested by Ware [10] to facilitate maximal visual differentiation by the user: red, blue, green, yellow, orange, brown, and purple. To support answering various research questions, the user is able to modify the visualization. Firstly, when the user hovers a vertex all individual edges of the corresponding editions are drawn in the dedicated colors. This mean of interaction helps to highlight the paths containing the dedicated token and to clarify those editions forming majority edges. Secondly, unimportant editions can be removed from the graph. Thirdly, the user is able to select one of the editions as a main branch, so that the corresponding vertices are drawn on the same horizontal level – variations to the other editions can be considered easily.

During the development phase, the humanities scholars of our project steadily evaluated our design, so that the result remains intuitive even for the inexperienced, maybe sceptical user. We strongly recommend such an iterative process when developing visualizations for humanities applications as it turned out to be very successful. In comparison to a plain graph layout, the presented design for the Text Variant Graph and the project page reminds the user, that it is a book to be read, not just some string of letters – which was a major concern of the humanities scholars.

Our presented approach is still applicable for examples where whole blocks of text have different orders among the various editions, but matching text blocks may strongly drift apart. In the future, we direct our attention on developing algorithms that visually align such structures more properly.

References

1. Desmond Schmidt and Robert Colomb. A data structure for representing multi-version texts online. *International Journal of Human-Computer Studies*, 67(6):497 – 514, 2009.
2. Ronald H. Dekker and Gregor Middell. Computer-Supported Collation with CollateX: Managing Textual Variance in an Environment with Varying Requirements. *Supporting Digital Humanities 2011*, University of Copenhagen, Denmark, 17–18 November 2011.
3. Tara L. Andrews and Caroline Mac. Beyond the tree of texts: Building an empirical model of scribal variation through graph analysis of texts and stemmata. *Literary and Linguistic Computing*, 2013.
4. Multiversiondocs: Merge, edit and compare N versions in one document. `http://code.google.com/p/multiversiondocs/` (Retrieved 2013-10-30).
5. Martin Wattenberg and Fernanda B. Viégas. The Word Tree, an Interactive Visual Concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, November 2008.
6. Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for Visual Understanding of Hierarchical System Structures. *Systems, Man and Cybernetics, IEEE Transactions on*, 11(2):109–125, Feb 1981.
7. Stemmaweb – a collection of tools for analysis of collated texts. `http://byzantini.st/stemmaweb/` (Retrieved 2013-10-30).

8. TAdER – Text Adaptability is Essential for Reading. http://www.tader.info/scrolling.html (Retrieved 2013-10-30).
9. Holy Bible – Verses in Various English Translations. http://informatik.uni-leipzig.de/HolyBible (2013).
10. Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 3rd edition, 2004.