Explaining Semi-Supervised Text Alignment through Visualization

Christofer Meinecke, David Joseph Wrisley, and Stefan Jänicke

Abstract—The analysis of variance in complex text traditions is an arduous task when carried out manually. Text alignment algorithms provide domain experts with a robust alternative to such repetitive tasks. Existing white-box approaches allow the digital humanities to establish syntax-based metrics taking into account the spelling, morphology and order of words. However, they produce limited results, as semantic meanings are typically not taken into account. Our interdisciplinary collaboration between visualization and digital humanities combined a semi-supervised text alignment approach based on word embeddings that take not only syntactic but also semantic text features into account, thereby improving the overall quality of the alignment. In our collaboration, we developed different visual interfaces that communicate the word distribution in high-dimensional vector space generated by the underlying neural network for increased transparency, assessment of the tool's reliability and overall improved hypothesis generation. We further offer visual means to enable the expert reader to feed domain knowledge into the system at multiple levels with the aim of improving both the product and the process of text alignment. This ultimately illustrates how visualization can engage with and augment complex modes of reading in the humanities.

Index Terms—Text Alignment, Word Embeddings, Human-in-the-loop, Visualization in the Humanities, Professional Reading.

1 Introduction

Medieval vernacular literary texts often exist in multiple versions that are characterized by significant differences in length and structure. This textual instability is known as *mouvance* [1] and takes on a wide variety of forms: differences in regional or scribal dialect, influences of an oral tradition, as well as the poetic modification of wording, rewriting, even omission or rearrangement of large parts of the text. These unique properties of literature pose a challenge when analyzing different versions of a text manually. The principle aim of the visual analysis of *mouvance* is to generate new perspectives that allow expert readers to draw conclusions on dependencies across the different versions and language dialects, as well as to track, compare and assess the use of language, its meanings and time-dependent changes. The precondition for such analysis is discovering similar text fragments across different text versions, a technique known as *text alignment* [2]. For carrying out the alignment of complex texts marked by *mouvance*, we determine similarity at the level of lines (verses or sentences). Figure 1 illustrates an example of alignment of two versions of the medieval French epic poem, the *Song of Roland*. Colored streams connect lines of the two versions that share a certain degree of similarity.

Straightforward solutions to determine accurate text alignments do not exist for medieval vernacular literary texts. As a first solution, we proposed the visual analytics system *iteal* [3] for interactive visual comparison of complex text versions in support of professional reading [4], for researchers that we call here "expert readers". This user-driven, parameter-based, white-box system—henceforth *iteal-V1*—uses string similarity and word n-grams in order to align and visualize different versions of a text. Although *iteal-V1* provides expert readers with a transparent framework to study differences and similarities across different text versions, its major shortcoming lies in the neglect of semantic text features like words with inflectional endings, synonyms at the word level and stylistic features formed by the combination of words such as paraphrases or analogies. As taking into account such features which determine alignments across text versions is crucial to produce an optimal result, we replaced the white-box text alignment computation back end with an unsupervised word embedding method [5] to accommodate semantic alignments. This second version of *iteal*—henceforth *iteal-V2*—is a fully automated text alignment approach. However, since expert readers are not typically specialists in machine learning or advanced natural language processing, implementing such pipelines for domain-specific problems without providing a means to understand or interact with the results can be problematic [6]. Furthermore, expert readers would like to be able to observe, evaluate and critique such automated processes, and they are increasingly interested in peeking into computational black boxes to understand their assumptions and their inner logic [7], [8].

With such a critical, interpretative expert reader in mind, this paper describes a series of extensions in a version we call *iteal-V3*. Both *iteal-V1* and *iteal-V2* are limited in as much as they do not incorporate expert reader domain knowledge into the calculation of textual alignments. Given the strict rules of *iteal-V1*, we instead directed our attention towards generating a method to allow the expert reader to adjust the word embeddings, with the effect that text alignment results also change in an iterative manner. We offer feedback

- *Christofer Meinecke is with Leipzig University. E-mail: cmeinecke@informatik.uni-leipzig.de.*
- *David Joseph Wrisley is with New York University Abu Dhabi. E-mail: djw12@nyu.edu.*
- *Stefan Jänicke is with University of Southern Denmark. E-mail: stjaenicke@imada.sdu.dk.*

Manuscript received April 19, 2005; revised August 26, 2015.

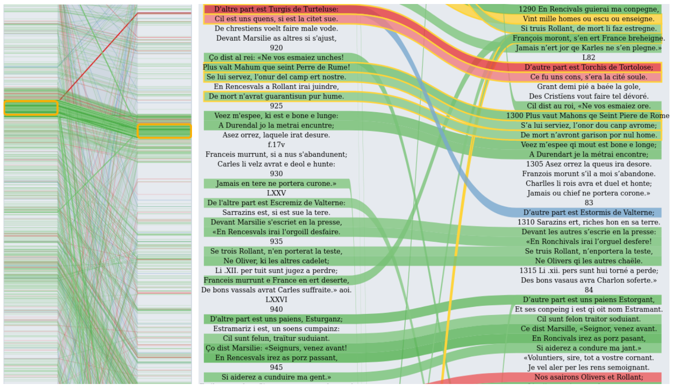

Fig. 1: A barcode and a side-by-side view of two versions of the *Song of Roland* show different types of alignments.

mechanisms and novel, manifold perspectives on alignment and word relations, with a particular eye for their legibility. *iteal-V3* can be used to explore the neighborhood of lines and words in the vector space, and simultaneously to provide insights on step-wise generated results. We needed to develop new visualizations in order to match semantically closer concepts with the word embeddings and also to explain their behavior to the expert reader/collaborator, whom we refer to below as *DJW*. Over multiple iterations, the expert reader cannot only observe the changes in the alignments until a satisfactory result is obtained but can also assess the impact of human input on two levels: the alignment of the poetry as well as the adjustment of the words in the vector space. Our system makes the argument that the alignment of complex poetic tradition is not a linear process, but rather an iterative one based on cooperation between the model and expert reader.

Continuing our longstanding interdisciplinary collaboration [3], [5], [9], we adopted a participatory design process, proven to be valuable in designing visualizations to be understood and used by domain experts [10]. In summary, the contributions of this process to the community are:

- **Semi-automated Visual Text Alignment:** We provide an interactive, semi-automated text alignment approach, combining visual analytics, word embeddings and an iterative refinement process.
- **Visualizations for Word Transportation:** We designed visualizations to explain the computation of the Word Movers Distance [11], the distance measurement of our word vector approach.
- **Visualizations for Word Vector Neighborhood:** We introduce new visual means to observe, interact with and manipulate word vectors. Manipulations of word vectors affect the alignment results in our system, which make it important to allow the expert reader to trace changes in both word vectors and their neighborhoods across different iterations.
- **Reflection on the Participatory Design Process:** We document our design process that includes iteration-dependent reflection on how the underlying word embedding and potential changes were perceived, and what visual cues were required to better understand the alignment computation.

2 Related Work

Our work combines three different lines of inquiry. First, we focus primarily on the visualization of text variations and text reuse on a line-level alignment. Second, we design multiple visualizations, which focus on the relation of the k-nearest neighbors of a word vector of interest. Third, through the interactions with the model, we engage with research focusing on active learning, and related works applying a human-in-the-loop scenario to include domain knowledge for textual analysis. The following subsections are dedicated to those three aspects.

2.1 Visualization of Textual Variance

Text variants are important elements of different domain-related tasks [2]. For example, applications have been developed to support analyzing patterns of text reuse [12], [13], [14], and more particular, plagiarised text fragments [15]. Furthermore, text variants appear in different languages and visualizing automatically aligned fragments can assist translators in manually adjusting them [16].

However, most applications focus on different versions of a base text, as in our scenario. Asokarajan et al. [17] tailored their system to support analyzing lemma-level similarity for classical Latin texts. Other systems focus on directly comparing two different versions of a text [18], [19], [20]. Some systems do not apply text similarity measurements and use manually collected annotation features like Baumann et al. [21] to compare two critical editions. In order to compare different translations of Shakespeare's Othello, ShakerVis [22] uses a vector space model and applies a parallel coordinates plot and scatter plots to analyze occurring patterns, while Alharbi et al. [23] visualize alignments in parallel translations through stream graphs. Hazem et al. [24] align medieval devotional text editions using different methods, including pre-trained word embeddings and visualize text similarity in a heat map.

2.2 Visualizing the Nearest Neighbors of Word Vectors

Modern natural language processing pipelines often apply dense word vectors as a representation of words. This shift from sparse one-hot encoding to dense word vectors was brought upon by Mikolov et al.'s word2vec [25], [26], [27]. Despite the wide application of word embedding models, only a few works visualize the vectors and their relations. Most of them apply dimensional reduction through PCA, t-SNE, or UMAP to project the vectors to a 2D or 3D space and then visualize them as a scatterplot like the Embedding Projector [28], WebVectors [29], UTOPIAN [30], DataDebugger [31], or ConceptVector [32]. In contrast to these works, we do not primarily focus on dimensional reduction. Instead, we simplify nearest neighbor graphs [33], [34] by offering a one-dimensional representation of word neighborhoods to make the constitution of the vector space comprehensible to the expert reader. Similar to most of the above-mentioned related works, we allow the inspection of the nearest neighbors of a word of interest, but we go beyond inspection, also allowing the original vector space to be changed through interaction. Changes in the neighborhood relation are further visualized after such interactions.

Similar interactive methods based on word vectors were applied by Park et al. [32] to interactively construct lexicon-based concepts or to refine topic models [30], [35].

2.3 Human-in-the-Loop for Text Analysis

Machine learning methods are found in many domains, often without incorporating user feedback or visualizations that explain their functionality. By consequence, a large number of systems and architectures are designed as de facto black boxes. This is particularly acute for deep learning models [36]. In recent years, various methods and concepts have been introduced to tackle the opacity of such black box systems [37], [38] in order to give users of these systems ways to understand them, to interact with them, even to critique their performance. The application of user interaction as feedback to a model is indicative of a human-in-the-loop process, in which a model is iteratively refined. The question remains, however, how users of a system are able to assess the step-wise refinement.

A popular concept for model refinement is active learning, which is applied when manual data labeling is impracticable. A user labels data samples that are chosen by the system to maximize the feedback and to minimize the training time. In some cases, this process is combined with interactive visualizations to better understand the classifier [39], [40], [41]. Snyder et al. [41] combine interactive visualization with active learning for the classification of streaming text data, while Heimerl et al. [40] compare different strategies for active learning of document classifications. Furthermore, Kucher et al. [39] apply active learning for manual text annotation and visualize the annotations based on the corresponding categories. Maki et al. [42] apply active learning to construct context-specific sentiment lexicons with minimal user interaction.

3 Project Overview

Our interdisciplinary collaboration began in 2015. With a corpus of medieval poetry at hand, our mission was to develop a visual analytics framework capable of discovering aligned text fragments taking into account the expert reader's domain knowledge about the phenomenon of *mouvance* [9]. In 2017, we published *iteal-V1* [3] which determines alignments based on string similarity and shared word n-grams. Although string similarity can disambiguate many of the medieval French words, the limitation of this approach is its inability to take into account semantic features characteristic of vernacular, orally-influenced poetry, such as synonymic replacement, formulaic intertextuality, word reorganization or significant orthographic difference. To address these problems, we proposed *iteal-V2*, an automated approach based on word embeddings [5].

This work extends the *iteal* portfolio with *iteal-V3*, introducing novel visual metaphors to communicate the shape of the vector space, word neighborhoods, and iterative changes introduced into the vector space. Following Munzner's guidelines for task abstraction [43] the domain-specific tasks for all *iteal* versions are to *derive* alignments from lines in the poem that can then be *explored* by the expert reader who then *identifies* alignments of interest and *annotates* them as true or false alignments. *iteal-V3*'s visualizations serve the need to understand and to adjust the word embeddings used for alignment computation. What follows is a description of the text corpus and details of how the word embeddings are computed. The whole process is summarized in Figure 2. We tested our system for medieval French epic poetry, but the pipeline is applicable to other languages and generalizable to other corpora with a high degree of intertextuality [44].

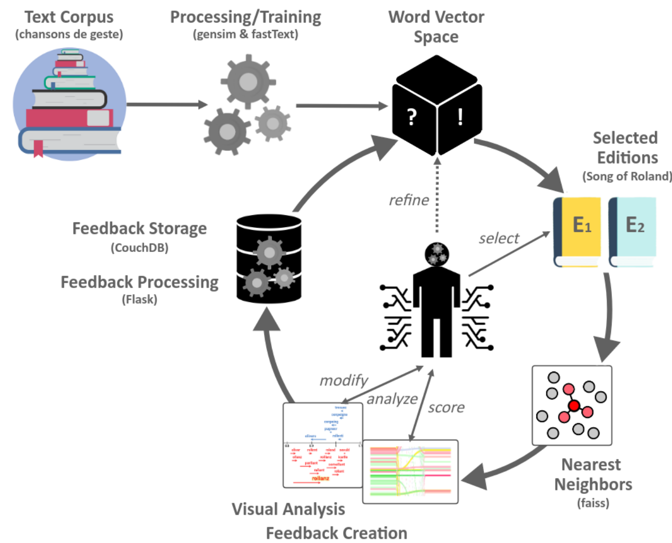

Fig. 2: Our human-in-the-loop process.

Text Corpus and Pre-Processing. Our historical text corpus consists of multiple medieval French poetic works, in the epic genre known as *chansons de geste* along with some texts of the Alexander legend which share epic-like characteristics. The corpus varies in terms of language variety, epic cycle and century and consists of around 30 different works. Our alignment here focuses on the oldest of the epic legends and arguably the most complex, the *Song of Roland*. The Roland tradition was chosen due to its significant intertextuality and variance. For example, different versions of the *Song of Roland* can vary from 2000 to 8000 lines long. Shared narrative aspects of the Roland tradition across the different versions make the exercise of comparing them a compelling task. We focused on the alignment of texts taken from single-manuscript editions of the *Song of Roland*: the Oxford manuscript (about 4,000 lines) and the Venice 7 manuscript (about 8,000 lines). The manuscripts are written in major regional varieties of medieval French and this variety adds another layer of complexity to the alignment. The whole corpus was cleaned from diacritics (editorial emendations not present in medieval language), unnecessary white spaces, and artifacts created through Optical Character Recognition.

Word Embeddings and Post-Processing. A pre-trained model for modern French was available, and so we began with carrying out an alignment of a structurally conservative modern translation of the Oxford version (Petit de Julleville) [45] with the original medieval text. DJW evaluated the results as arbitrary, on account of the wide gap between the medieval and modern French languages. Consequently, we had to use a model for twelfth-century French for which no solutions exist. Therefore, we trained a model based on the text corpus described above using the *gensim fastText Skip-Gram* implementation [46], introduced in *iteal-V2*. We applied *fastText* [47], owing to its capability to grasp orthographic variance of different dialects and word modifications over time, thus addressing the issue of highly variant spellings. An evaluation of the different approaches can be found in the Supplemental Material Section A. Because of the small corpus size, we decided on a 100-dimensional vector space for computing the word embeddings. After the training phase, word vectors were normalized and post-processed. A subsequent normalization ensured unit length and improved the quality of the word vectors, since vector length is known to correlate with word frequency [48], that is, in our case when dealing with rare orthographic variants not important for the meaning of a word. For the post-processing step, the method of Mu and Viswanath [49] is applied to eliminate the common mean vector and the top dominating direction of the word vectors. This leads to more uniformly distributed vectors, which can help in better expressing the word similarities and further reduce the influence of the word frequencies on the vectors [48].

Compute Alignment Candidates. When comparing two text versions, a sentence vector is computed for each line using unsupervised Smooth Inverse Frequency [50]. With *faiss* [51] these sentence vectors are added to an index structure to query the nearest neighbors of each sentence based on cosine similarity. For each text version, an index is constructed and the other version is queried. This process results in a list of potential alignment candidates for each line in both versions, thereby reducing the computation time for the following steps. For each candidate, two sentences X and Y, the Word Movers Distance (WMD) [11] is computed by solving an optimization problem to find the minimal cumulative Euclidean distance between the sentences' word vectors, i.e., the minimal cost required to transport sentence X to sentence Y. We denote the set of transportation pairs of two sentences X and Y as $T_{X,Y}$. A word transportation pair is a tuple $(w_1, w_2) \in T_{X,Y} : w_1 \in X \wedge w_2 \in Y$, while a sentence X is a bag of words $X = \{w_0, ...w_n\}$. We applied the WMD because it performs well for nearest neighbor classification [11]. In addition, the underlying word transportation metaphor can be easily visualized and interpreted. The resulting list of nearest neighbors for each line is used as input for the visualization system.

Need for Refinement. A visual analytics system facilitating the study of variant text traditions needs to address multiple usage scenarios as well as the means of visualization and interaction for a user-driven process of gaining insight. Such a process might include automatically detecting alignments, assessing the quality of these alignments, removing false positives, as well as adding new undetected alignments. Upon switching the alignment detection process to a word embedding model in *iteal-V2*, new scenarios for the expert reader indeed emerged, but the removal of the parameter adjustment opportunity of *iteal-V1* made it difficult to interpret the results. In particular, it was not traceable why the system aligns specific lines drawing on the underlying word embedding model. Traceability had been granted in *iteal-V1* by the string similarity approach, but the change to a word embedding solution did not allow us to incorporate user feedback for improving the overall alignment result. Thus, for *iteal-V3* we conducted an iterative process allowing the user-led refinement of the model to proceed in iterations (stages). To communicate line and word-level changes after each iteration, a variety of visualizations and interaction techniques were developed for evaluating line alignments and word vector relations as well as observing stage-dependent modifications.

Participatory Visualization Design. Bearing on the authors' experiences gained in a variety of interdisciplinary digital research, we followed a participatory visual design process to carry out this project [10]. This process builds on, but also extends, task-based development models [52], as most of the design considerations and adaptations were debated in-depth among all project members [53]. Our stage-based visualization development led to vibrant reflections on required adjustments on the one hand, but, more importantly, to entirely new visual perspectives on data and alignments on the other.

4 Iterative Design of *iteal-V3*

Our participatory design process started with a prototype, which allowed DJW to compare the results generated by *iteal-V1* [3] to the ones of *iteal-V2* [5]. The prototype offered an alignment view that allowed the introduction of user-generated input into the semi-automated system by scoring line-level alignments according to their reliability. In later stages, we added visualizations to explore the neighborhood of the word vectors and to allow for word-level modifications to the vector space. After each session, user feedback on line and word level is used to score line-level alignments among the text versions. Changes to the word embeddings can be inspected by the expert reader in the subsequent session. *iteal-V3* can be also applied to other alignment scenarios of two text versions provided that both a list of potential alignment candidates and the embedding model used to compute them are available.

Through multiple iterations, we developed a series of visualizations to inspect stage-dependent alignment changes and word embedding features. In what follows, we describe the visual encoding and means of interaction which we designed to engage with complex questions in the human textual record and the workflows of the expert reader.

4.1 Alignment View

DJW wants to inspect the results of the alignment of the Oxford and Venice manuscript using the unsupervised word embedding method of iteal-V2. In the beginning, he sees the barcode view showing a zoomed out version of the poem and the alignments, which he can use to jump to a specific area of interest in the poem. Next to it, he can see the side-by-side view, which allows him to read the editions while exploring alignments. For each line in the Oxford manuscript, the first nearest neighbor in the Venice manuscript is used for the alignment. Currently, the first stage is selected, which is the model after training and without any user interactions. DJW can later switch to a higher stage to see the influence of his interactions with the system. To reduce visual clutter and to focus on highly similar alignments DJW can increase the similarity threshold, which is by default the average similarity value.

Tasks. The alignment view is designed to support exploring alignments computed by the word embedding approach. It makes alignment patterns in a barcode and a side-by-side view visible, and it aids to identify particular alignment tuples that attract the expert reader's attention. A coloring scheme is implemented to facilitate identification of stage-dependent changes of alignment patterns.

Design. The parameter-driven *iteal-V1* system offered diverse visual means to inspect alignments and to show changes after parameter changes. For *iteal-V3*, the original alignment view was extended to communicate stage-dependent modifications as well as feedback information in a more dynamic fashion. A sample output of the barcodes and the side-by-side views both displaying alignments as colored lines can be seen in Figure 1.

The expert reader can interactively change the set of displayed alignments through the use of different sliders. The stage slider allows for inspection of different iterations (after feedback) of the model and the differences between them. The nearest neighbor slider determines the number of nearest neighbors that are displayed for each line. Since the neighbor relation is not symmetric the expert reader has the option to change if the neighbors of the first text, the second text or both are displayed. Furthermore, the similarity threshold can be increased to allow only inspecting high-quality alignments. We denote the alignments of stage i and the selected number of nearest neighbors k of two editions E_1 and E_2 as $A_{i,k} = \{(X,Y) : X \in E_1 \wedge Y \in E_2\}$. We additionally denote the alignments found in the current stage as A_c and the alignments found in the previously selected stage as A_p.

In the first stage, all alignments are displayed as green streams connecting two lines, one of each text version. For higher stages, alignments are grouped in different sets and color-coded. The set $A_g = A_c \cap A_p$ includes green-colored alignments found in both the current and the previously selected stage. The set $A_r = A_c \setminus A_p$ stands for new, red-colored alignments that were not found in the previous stage. Finally, the set $A_b = A_p \setminus A_c$ stands for blue-colored alignments that were found in the previous but not in the current stage. The system permits enabling or disabling the different alignment sets to focus on nearest neighbor relations of interest, and the quality of the alignment is communicated through the saturation of the line. If the feedback option is enabled, all alignments scored in the previous stages are visualized as yellow lines in the barcode view. In the side-by-side view, alignments already contained in any of the sets A_g, A_r or A_b receive a yellow border, otherwise, since they have been manually annotated, they appear in yellow as well.

For reference, the previously selected stage is used. If no stage has been selected, the first stage is used as a fallback to show the total changes from the beginning of the feedback process. If the similarity threshold is increased, alignments that no longer match the new value appear in grey but keep a colored border. Both the barcode and the side-by-side views are interactive allowing for a flexible exploration of the alignment space through scrolling or clicking on a text section of interest. If an alignment is selected, the line similarity view pops up.

Usage Scenario. At first glance, it seemed to DJW that *iteal-V3* offered less control of the visualization than previous iterations but in reality, it changed both the process and the kinds of alignment possible and foregrounded the idea of alignment as a gradual process. With the changes in functionality and the color-coding of step-wise reading, the new system was actually more effective in arriving at high-quality, nuanced alignments. Furthermore, the kinds of alignments that were found automatically were of a different nature. They resembled the broken n-grams and orthographic variance which had been identified previously, in addition to new kinds: lines of structural similarity, even lines sharing repeated formulaic speech or synonymous meanings. The alignment example depicted in Figure 3a illustrates how the shared string "seint Michel" referring to the feast day of Saint Michael is identified in the lines, but in addition, the co-presence of synonyms "feste" (feast day) and "jor" (day) contribute to the alignment of the lines. Whereas an expert reader might not at first recognize this phenomenon as a strong alignment, the semantic and structural relations that emerged on account of proximity in vector space provided unexpected, yet positive, suggestions for expanding the notion of intertextuality.

4.2 Line Similarity View

Now, DJW wants to inspect one alignment of interest in the side-by-side view. When he clicks on the corresponding stream, the line similarity view pops up. At the top, he can see the word transportation that is used to compute the WMD, the system's underlying similarity measurement. At the bottom, he can see a heat map showing the distance between the word vectors in both lines, their nearest neighbors, and, again, the word transportation. In a higher stage, he could also see the new and the previous similarity value of the alignment as well as the score that is saved in the database for the alignment. He decides to score the alignment and to save the score to the database in order to include this scoring feedback when computing the next iteration.

Tasks. The major purpose of the line similarity view is to provide visual explanation of the functioning of the WMD in the automated alignment computation that leads to showing this particular alignment tuple. If desired, the expert reader can include domain knowledge into the model by annotating the chosen alignment with a qualitative score.

Design. The two feedback visualizations for inspecting an alignment of interest can be seen in Figure 3. The first visual depiction (Figure 3a) conveys the word transportation of the WMD by saturated green arrows originating from words of the first sentence to words of the second sentence. The color scale ranges from white to green to show how much of a word has been moved to the connected word. In addition, a heat map shows the distance of the word vectors for the words in both sentences (Figure 3b). A high saturation indicates a lower distance following a linear color scale from green to white. In this view, word transportation is communicated through a solid border while the nearest neighbors of each word are displayed as striped squares. The heat map gives a quick overview of the distances among the observed words in the vector space. Both visualizations help in getting a feeling for word transportation, that is, why the corresponding lines are considered similar in the vector

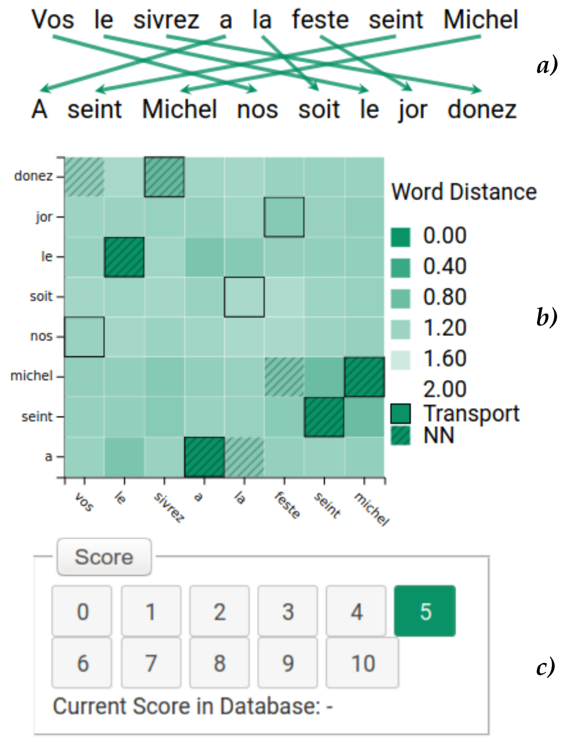

Fig. 3: The Line Similarity view allows to analyze the word transportation and the word similarities of an alignment of interest and to rate it.

space. Lastly, the line similarity view provides a scoring scale (Figure 3c) to be used to rate the alignment on a scale of 0 to 10; the scoring feedback is one of two possible means for the expert reader to refine the vector space for the next stage. Another way to score the nearest neighbors is to click on a particular line of interest in the side-by-side view. The expert reader is then presented with a list of nearest neighbors with the associated similarity values and scoring scales. The popup also lists the nearest neighbor of the chosen line from the previous stage to observe changes induced by the expert reader's feedback. To further investigate word vector neighborhoods, the Word Vector Space View can be opened.

Usage Scenario. Assessing the alignment of poetry using a numerical scoring scale is not a straightforward task, so DJW judged the relative quality of the alignment of two lines using different features that can occur together in the same line. Generally speaking, an alignment received a 10 if the linguistic information was exactly the same, even if the words in the lines were spelled very differently, or close synonyms were used and verbs were found in different tenses. A score of 0 indicated that there were no shared words or shared semantics between the lines. Using the ten possible intervals, DJW would score based on the amount of shared linguistic information in the line, relative to the number of words or syllables. For example, a score of 5 was attributed to the sentence in Figure 3a since a little more than half of the words in the line are similar, with two important differences, one was a synonym and the other the definite article of another gender.

4.3 Word Vector Space View

DJW decides to investigate the neighborhood of the words included in the aligned lines in the word vector space. He can select a particular word and a number k of nearest neighbors that are shown in the word vector space view. He can move the words closer or farther away from the target word in order to change the underlying vectors, which are used for the computation of the alignments in the next iteration. If DJW wonders why two words $\vec{a}$ and $\vec{b}$ do not match as pairs for the computation of the WMD, he can make use of the neighborhood intersection view for exploring the situation. There he sees the common neighborhood of $\vec{a}$ and $\vec{b}$ and all words that are closer to $\vec{a}$ and, respectively, $\vec{b}$, than the target words are to each other. Based on the results, he can decide to change the distance between some of the word vectors. After scoring alignments in the line similarity view and moving words in the word vector space view, he can submit the collective feedback and trigger a re-computation of the word embedding displayed in the next stage.

Tasks. Our word space visualizations provide a simplified depiction of the neighborhood of words in order for the expert reader to understand line-level alignment decisions more easily. Whereas the word space view makes the neighborhood of a single word explorable, the neighborhood intersection view makes the neighborhoods of two words comparable. The expert reader is further able to conduct word-level adjustments by decreasing or increasing the distance between words.

Word Space View. In the word space view, the k-nearest neighbors of a word of interest are displayed on the x-axis as seen in Figure 4a. To prevent overlap of words, a collision detection is used to adjust their y-coordinates. This view gives an intuitive overview of the neighborhood of a word vector and allows the expert reader to change the distance d between two word vectors $\vec{a}$ and $\vec{b}$. One or multiple words can be selected and moved to a new position via drag and drop, especially if the expert reader concludes that their distance is inaccurate and two words should be either closer to each other (or more distant) in the vector space. To support this task, sample sentences giving the word in context can be observed in a popup. For each adjusted word, the new x-coordinate is used in the next iteration to adjust the word vector corresponding to the new distance d'.

$$\vec{a}' = \frac{\vec{a} + (\vec{b} - \vec{a}) \cdot (1 - \frac{d'}{d})}{\|\vec{a} + (\vec{b} - \vec{a}) \cdot (1 - \frac{d'}{d})\|}$$

The vector $\vec{a}$ of the moved word is moved closer to, or farther away from, the target vector $\vec{b}$ on the line between them. Finally, the vector is normalized to ensure unit length, resulting in a little inaccuracy in the distance d'. Through this visualization the expert reader can adjust the distances between words of interest, thereby changing their vectors. This approach could prove to be helpful when applying word embeddings to under-resourced languages in tackling training limitations.

Neighborhood Intersection View. In order to enable a more in-depth analysis of how the vector space is composed and to illustrate the relation among a set of word vectors, we provide a means for visual comparison of the neighborhoods of two word vectors. This is especially helpful if the scholar wants to investigate why synonymous words

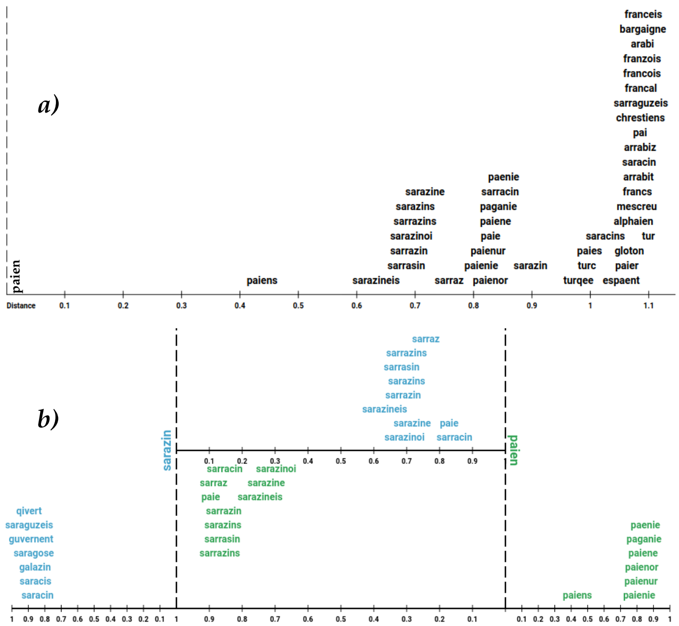

Fig. 4: The word space and the neighborhood intersection view visualize the nearest neighbor relation of words of interest.

are placed far from each other, or if they are transported to less related words. The visualization can be seen as a one-dimensional projection of a high-dimensional sphere with a diameter d being the Euclidean distance between the selected words $\vec{a}$ and $\vec{b}$. The two neighborhoods of $\vec{a}$ and $\vec{b}$, which are placed along two vertical axes as outlined in Figure 4b, are laid out in three sections of the screen. Words that are inside the sphere, thus having a distance to both $\vec{a}$ and $\vec{b}$ smaller than d, are placed in-between the vertical axes reflecting their orientation inside the sphere. Words outside the sphere with a distance smaller than d to either $\vec{a}$ or $\vec{b}$ are placed to the left and to the right of the vertical axes. Since the neighbor relation of the vectors is not symmetric, the words are color-coded to highlight which neighbors belong to which word of interest. For one specific alignment, all combinations of words in both sentences can be investigated. Two words can be chosen arbitrarily for comparison as well.

Usage Scenario. In addition to synonyms of any given word, we also found what might be called synonymous collocates. Taking the example of the word *paien* (pagan), DJW found in its immediate word space a number of prevailing, racialized stereotypes of the genre and the medieval period, words such as *Turc* (Turk) and *Sarrazin* (Saracen), as can be seen in Figure 4. Word embedding models, indeed, are particularly efficient ways of exposing cohesive discursive patterns of genres, “complete with [their] biases” [54], [55]. Notable in the word space was the appearance of other derived forms of the same words (nominal, adjectival or adverbial forms) that in the word space reflected a semantic cluster significant to medieval French epic. In Figure 4a, we find the nominal declension “paiens”, three derived nouns suggesting a place or state of being of the pagans, “paienie”, “paenie”, and “paganie”, as well as an analogously related group of words related to “sarrazin”.

Since the word space view in Figure 4a was designed for DJW to move words based on their similarity, a similar approach for evaluation was adopted as the one described above in Section 4.2; the same word, but with different spellings or inflections, was moved fully, whereas derived forms or synonyms were moved only a partial distance. Often we would find verbs of a totally different meaning, but in the same inflection (infinitives or 3rd person plural), usually from the end of the poetic line in rhyme word position. In addition, proper names, place names and ethnonyms tended to occur within the nearest neighbors, not on account of synonymity, but perhaps due to their common position within the prosody or syntax. This is the case especially of the words that would gather at the top left, such as “franceis” (French) or “arabi” (Arabs). As DJW became familiar with the neighborhood view for the word space, it became obvious how useful it could be as a cross-dialect workaround for grouping words from a common lemma or for building cross-dialect synonym lists perhaps to replace the problematic ones we have today [56].

4.4 Compare Stages

DJW can now investigate the second stage. In the alignment view, the alignments are now color-coded to show which alignments stayed the same and which were either removed or added as a result of his interaction. He can also focus on the alignments that he scored in the previous stage. Additionally, he can enable and disable the different alignment types. Furthermore, he can apply the word-level view to focus on the word vectors to find places in the poem where the vectors either changed significantly or not at all. He then can search for a specific word of interest and investigate its neighborhood. In the word space difference view, DJW can see the change of the nearest neighbors of a word of interest compared to the last stage. He can see which words were moved farther away and which moved closer. In a higher stage, he can investigate the changes over all iterations. With this input, he can repeat the scoring for alignments and words to further improve the model.

Tasks. All visualizations in this category are visual feedback mechanisms to reflect on how the expert reader’s adjustments have affected the model. They serve to compare word-level changes across stages, to identify words that have been impacted but have not been directly modified by the expert reader, and to open avenues for continued scoring activities.

Word-level View. In order to observe word vector changes, the expert reader can toggle from the alignment to a word-level view. Instead of alignments, words are now exposed using different colors. The word-level view serves two different purposes. Either the difference between a word’s vectors in two different stages or the change of a word’s neighborhood in the vector space is displayed. These changes are indicated by colored word backgrounds. The hue of the color depends on the amount of word vector change. To facilitate easy visual differentiation, colors are assigned to five sets of word moves based on the maximum observed change $d_{max} = max\{d(newV_w, oldV_w) : w \in V\}$, with d being the Euclidean distance between the new and old vector w, and V being the vocabulary of the text versions of interest. The colors assigned to the bins are: *none* for

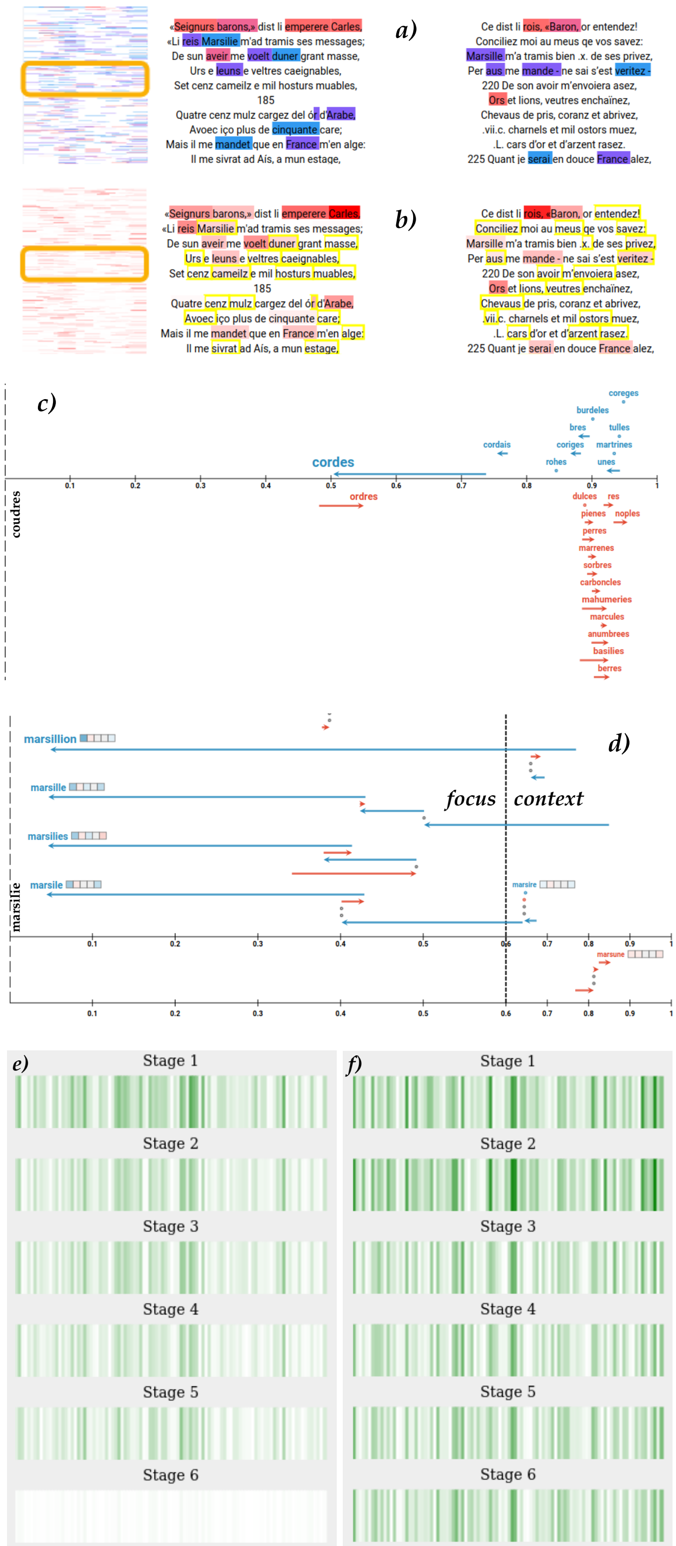

Fig. 5: The word-level view, which allows spotting places of interest, shows either the change of (a) a word vector or (b) a word's neighborhood. The word space difference view gives an overview of the changes in the vector space of a word of interest for (c) one or (d) multiple iterations. The dimension heat maps for each stage for the words (e) "marsilie" and "marsille" and for (f) "paien" and "sarazins". A higher saturation encodes a larger difference in the dimension. The Stage 6 heat map in (e) is almost completely white because of the low difference, which can also be seen in (d).

$d = 0$, i.e. if the vector has not changed at all, *blue* for d in $(0, \frac{d_{max}}{4}]$, *purple* for d in $(\frac{d_{max}}{4}, \frac{d_{max}}{2}]$, *pink* for d in $(\frac{d_{max}}{2}, \frac{3 \cdot d_{max}}{4}]$ and *red* for $d \geq \frac{3 \cdot d_{max}}{4}$. An example can be seen in Figure 5a. For the word neighborhood, saturation is used to encode the amount of neighborhood change similarly using a linear scale between white and red. An example can be seen in Figure 5b. We compute the change in the neighborhood for each word w as $\sum_{i=1}^{k} d(newV_{w_i}, oldV_{w_i})$ with k set to 50 and w_i being the i-th nearest neighbor of w. To prevent interferences among both channels, word moves and neighborhood changes are observed separately. As in the alignment view, feedback information can be superimposed using yellow color. This includes words manually moved in the previous iterations, which receive a yellow border in the side-by-side view. This helps to spot feedback interactions that may have had an impact on the vector space. When focusing on neighborhood changes, we use the metaphor of yellow borders to highlight words without vector changes. This supports finding words, not touched by any feedback interaction, with changed neighborhoods.

Word Space Difference. In order to observe how the neighborhood of a word has changed across two stages, either through directly moving words using the word space view or by alignment scoring, we designed the word space difference view. Similar to the word space view, the words are displayed based on their similarity to a target word. The difference is the inclusion of information of a previous state of the model. The changes are encoded as arrows starting at the old value and pointing to the new value as outlined in Figure 5c. In the case of minimal or no changes, circles are used instead. The words are separated into three groups, decreased distance, no change and increased distance. These three groups are stacked atop each other. Blue color encodes decreased distance, red color encodes increased distance and grey encodes no change. As per user preference, the font size of the word can either encode the absolute distance or the change in the distance. Although both are encoded also by the position of the word and by arrow length, this function is useful for guidance through the neighborhood.

Word Space Difference Sequence. We extended the word space difference view to communicate changes in the neighborhood of a word after multiple iterations, an example is shown in Figure 5d. For a reference word, multiple glyphs indicating distance change per stage are stacked. This information is further encoded in a heat map placed next to a word. The visualization gives a quick overview of how the neighborhood of a word of interest has changed for each iteration and throughout the whole process. We also used this view to observe the changes in the vector space after normalization and post-processing. To prevent scaling problems, we applied a focus+context metaphor. The focused part on the left-hand side shows the close neighborhood of a word while the context part on the right-hand side provides screen space for the remaining vector space. The expert reader can change the ratio of focus and context by dragging the separating vertical axis.

Dimension Heat Maps. Inspired by a barcode visualization for the comparison of word embeddings [57], we designed dimension heat maps to allow the expert reader to see the changes between two word vectors. An example of the words "marsilie" and "marsille", which are different variants of the same word can be seen in Figure 5e. Two words that are prevailing stereotypes of the medieval genre "paien" and "sarazins" can be seen in Figure 5f. Each dimension is encoded as a line ranging from white to green. A saturated green highlights larger differences, and white means no difference. Across five iterations, the difference between the word vectors is visibly reduced. The large change between Stage 5 and 6 relates to a word move in the last feedback session, in which DJW moved "marsille" very close to "marsilie", resulting in almost the same vector for both words. Something similar happened to the words "paien" and "sarazins". The visible difference between Stage 2 and 3 occurs as both words were moved by DJW closer to the word "sarrazins". For both word pairs, the changes in the other stages correspond to the more fine-granular alignment scoring process.

Usage Scenario. Although the process of aligning the poems is not brought to completion, with iterative scoring and word movement the system allows for a gradual convergence on strong patterns of intertextuality that were not identified by *iteal-V1* [3]. The challenge in such a complex, multi-stage task was to trace the impact of the changes DJW made. The color-coding scheme applied in Figure 5a was helpful not only for keeping track of the many words that were changed, but also, for purposes of coverage, to be able to redirect attention to sections of the poem or even sections of the poetic line (beginning, middle or end) for drawing DJW's attention to elements of alignment that may have been neglected in previous iterations. The same can be said of the saturation where DJW would pay more attention to words that were not yet affected by neighborhood changes. For example, in the list of animals given in the Venice poem in Figure 5b in lines 6 and 7, the neighborhoods of the names of the animals "ors et lions" (bears and lions), "veutres" (hunting dogs), and "chevaus" (horses) had moved but the various action verbs at the end of the line had not. Furthermore, for more precise indication of word space difference discussed in Section 4.3, word space difference and word space difference sequence are particularly helpful for more precise separation of words that often occur in the rhyme position, for example, "cordes" (the Spanish city, Cordoba) and "ordres" (order or position), yet without any semantic similarity. Visualizing the step-wise progression of the words that were affected by word moves with the dimension heat map, ultimately provides the expert reader interested in forms of complex intertextuality to focus not on a perfect alignment, but rather to self-pace and self-monitor while exploring complex textual scenarios with companion tools, exploring the relations between different phenomena at hand, and assessing the evidence on display [4].

4.5 Alignment Scoring

The most important feedback opportunity for DJW is scoring an alignment, dependent on its feasibility on a scale from 0 (entirely unreasonable) up to 10 (perfect match). In general, the scale indicates how similar the words in the alignment are, privileging syntactic as well as semantic features. Since it can be difficult to accept or reject alignments outright given the nature of the poetry at hand, DJW requested a means of being able to interpret the lines in more depth.

After scoring several alignments they are used as feedback to the word vectors. Our feedback approach is inspired by the Rocchio Algorithm [58], which moves a query vector closer to relevant documents and farther away from irrelevant documents. Our adjusted Rocchio formula results in:

$$\vec{v}_w = \alpha \cdot \vec{v}_w + \beta \cdot \frac{1}{|D_{p_w}|} \cdot \vec{v}_{p_w} - \gamma \cdot \frac{1}{|D_{n_w}|} \cdot \vec{v}_{n_w}.$$

In the classical formulation, D_p and D_n are sets of relevant and irrelevant documents. In our case, they correspond to bags of words, which should be closer to the target word w or farther away. In contrast to the Rocchio Algorithm, we do not focus on a query vector, instead, we apply an update for all word vectors in the scored alignments, which we separate into three bins: positive feedback (alignments scored higher than 6), negative feedback (scored lower than 4), and mixed-case (scored 4 to 6). We decided to apply this approach because of *hemistiche* (half-line) alignments. This is important across versions of a medieval poem since sometimes the information of one line is transposed into a single line in the target poem, and other times it is split in half across two separate lines. This can also be an issue when a poem is recast in a different meter and recombination of syllables or words is necessary. For a given word w, D_{n_w} includes, for all alignments (X,Y) with $w \in X$ or $w \in Y$, all words of the other sentence in the alignment if the score is lower than 4. Alignments with a low score are typically generated through overlapping function words or misplacement of rare words in the vector space, so it can be beneficial for the following iteration to move all words appearing in this false alignment slightly away from each other. Similarly, D_{p_w} includes all matches of the word transportation problem $T_{X,Y}$ for the word $w \in X$ or $w \in Y$ in all scored alignments (X, Y) with a score higher than 6. The case of a score between 4 and 6 corresponds to half-line alignments. Because the sentences are still not totally dissimilar, the transportation pairs are added to D_{p_w}, while all the other word pair combinations are added to D_{n_w}. This combines the positive with negative feedback to reduce the risk to move similar words away from each other. A more in-depth discussion and an example can be seen in Section B of the Supplemental Material. Another difference to the original formula is the computation of $\vec{v}_{p_w}$ and $\vec{v}_{n_w}$.

$$\vec{v}_{p_w} = \sum_{(t,s)\in D_{p_w}} \frac{s}{10} \cdot \vec{v}_t, \quad \vec{v}_{n_w} = \sum_{(t,s)\in D_{n_w}} (1 - \frac{s}{10}) \cdot \vec{v}_t$$

Instead of a simple centroid, we compute a weighted centroid based on the score s of the alignment (X, Y), in which the word w and t co-occurred. α, β and γ are weighting values to further control the influence of the original vector, the positive centroid and the negative centroid to the new vector. Modern information retrieval systems set $\alpha = 1$, $\beta = 0.8$ and $\gamma = 0.1$. We deviate from those default values and set $\beta = 0.5$ and $\gamma = 0.5$ to treat interactions of the expert reader equally. The scored alignments are stored together with the new distances of the moved words in the Word Space View. Both feedback types are applied to adjust the word vectors. The scored alignments are processed first, followed by new distances registered after word moves.

5 Evaluation

Our project setup profited from our longstanding collaboration and trust as a team, which allowed us to avoid the far-reaching misunderstandings typical of projects at the intersection of visualization and digital humanities [59], [60], [61], [62]. Our close collaboration also made possible the opportunity for an implicit evaluation of the co-designed iterative visual design process. Such evaluation has been documented before in various publications focusing on digital humanities applications [63], [64].

Iterative Scoring & Qualitative Evaluation. To begin, we focused on the *Song of Roland* as outlined in Section 3 and Section 4. In five sessions, DJW scored alignments and then moved words in the word space view. We met once a week to reflect on the results of each stage and to plan the steps for the next, revisiting the visualization design and incorporating DJW's feedback into the model. A detailed overview of the decisions in and after each iteration can be found in Section C of the Supplemental Material. The results from our extensive work with the *Song of Roland* have confirmed the extent of intertextuality in this tradition commented upon by generations of scholars, but never systematically and visually demonstrated. For an extended evaluation of our human-in-the-loop process, we carried out a sixth iteration on three different text traditions, each of which has multiple versions. First, we worked with two of the four "branches" of the *Romance of Alexander*, a term used to indicate different segments of the life of Alexander the Great compiled in medieval French [65]. These branches made up part of the training corpus for the initial model. Furthermore, we included two decasyllabic versions of the *Life of Saint Alexis* and two octosyllabic versions of the *Life of Mary the Egyptian*, which were not part of the training corpus and therefore contained out-of-vocabulary words that were added to the model by DJW through the scoring.

Quantitative Evaluation. To carry out a neutral assessment of whether our model suits its intended purpose, we sampled up to 60 sentences from each of the four text sources where the nearest neighbor had changed after the sixth scoring session. For each of these sentences, DJW was presented with a blind choice between the two nearest neighbors found before and after the sixth scoring session, and he had to rate which nearest neighbor is more accurate or if neither one is. We choose the sixth session because it was the first session during which DJW worked with all text sources. We have chosen to sample 60 sentences since for some of the text sources, there are not more samples for which a sufficient change in the vector space occurs. The results, summarized in Figure 6, document the gradual improvement of the model with our suggested methodology. For all text sources, more of the nearest neighbors determined after the sixth scoring session (green bars) were picked as the more accurate ones. The red bars show the lower numbers of cases in which DJW picked the nearest neighbor determined before the sixth scoring session as more accurate. The orange bars are cases in which neither of the two neighbors has been rated as more accurate.

Analysis. DJW further selected a reason for his evaluation based on the various features of alignment discussed above in Section 4. Because of the subjective nature of

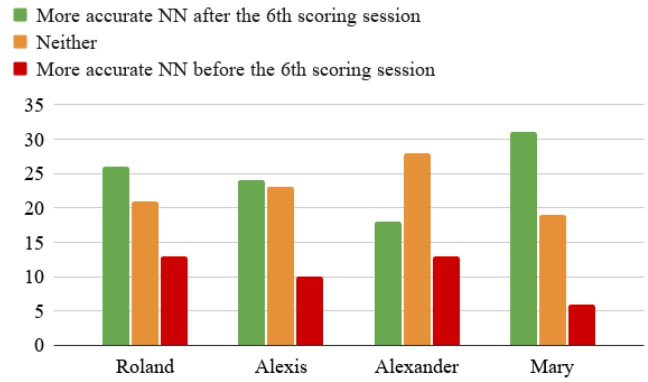

Fig. 6: Quantitative Evaluation Results

textual alignments between medieval text sources, we did not apply significance measurements for this process, instead, we focus on emphasizing their characteristics, while measuring them against an assessment based on content-specific knowledge. Since the branches correspond roughly to temporal segments of Alexander's life instead of rewritten versions of the same epic cycle, the lowest scores from the group were expected from these branches where *iteal-V3* was able to find only examples of similar poetic lines across the tradition. Most of the improvement of the model stemmed from the other three text traditions. It is important to note that the reasons for alignment were also not equally distributed across the various text sources, but seem to correspond to the nature of the orally-inflected texts in question. It was the choice of a differently inflected verb or noun that led to a new nearest neighbor to be chosen in the case of the *Life of Saint Alexis*. In the case of the Alexander romance, it was the choice of synonymic features that led to a previous nearest neighbor to be chosen, whereas in *Life of Mary the Egyptian* it was the deciding factor for a new nearest neighbor. In the end, it was orthographic difference that was the most dominant factor in the choice of both new and old nearest neighbors in the case of the *Song of Roland*. These data not only reflect the relative narrative similarity of parts of the versions of the Roland but also their significant dialectal differences. Notably, even though the *Song of Roland* was not touched in the sixth scoring session, it exhibited 26 out of 39 new alignments that were better than the previous model.

6 Discussion

Our development of a human-in-the-loop process drew upon an intense interdisciplinary exchange, from which we gained valuable insights for our respective scholarly backgrounds. Further, it led us to assess limitations of our approach and to discover directions for future research.

VIS Reflections. During the participatory design process and during the scoring sessions we logged DJW's interaction with the different visualizations. In the beginning, he mostly focused on line alignments and their scoring. The initial cautious interactions with the Word Space View changed over the course of the scoring sessions so that in the end the word interactions captured his attention more than scoring lines. Over time the interaction with the different options to enable or disable the different sets of alignments also increased. In the whole process, 525 different alignments were scored and around 770 word vectors were manually moved. The rating of the alignments involved alignment interactions, observing an alignment of interest and its associated word transportation visualization, as well as direct interaction with the line to inspect and rate a list of nearest neighbors of the sentence. The Word Space View was typically accessed from the Line Similarity View, and the direct search for a word of interest was seldom used. A reason for this behavior can be observed when looking at the feedback in the database: in later stages, the scoring of an alignment was often combined with a Word Space interaction for the words appearing in these alignments as DJW was moving through the poem.

DJW Reflections. From the point of view of the researcher using the system, it is a rather complex environment and its complexity has both benefits and drawbacks. To begin with the latter, the learning curve with such a system can be steep, not because its visual semantics are unclear, but rather because they are so precise and interconnected. Learning to read efficiently within such an environment can take time, and in particular, with learning to integrate high-level observations from the vector space into more granular reading practices. The researcher has to become accustomed to navigating the various decision making and presentational views of *iteal-V3* and to assess how, or if, they can integrate such data into decision making. On the positive side, it is possible to have rather complex interactions with poetic texts, to compare them in novel ways, and in so doing, refine the word space of the genre in question. Debates about multi-scalar reading in recent years have uncovered rich examples of distant reading practices in "a long view of disciplinary history" [66]. Scholars also have asserted that close and distant reading are not opposites [67] and a loose consensus has emerged in the critical literature that not only interpretation at different scales is a valuable contribution to contemporary literary studies, but also that visualization has a key role to play in facilitating such innovative reading practices [68]. *iteal-V3*'s visual system does not blend all aspects of close or distant reading–it would be absurd to claim that it did–but it does combine a very specific task of professional reading, the process of synoptic comparison of texts in view of understanding textual genetics, with additional views of more abstract representations of the word space of the genre. In *iteal-V3*, visualization is not only the static output of computational analysis of texts, as unfortunately is the case in much literary historical criticism, but instead, it forms an environment in which active interaction between close, distant, and other acts of reading that fall in between, might take place.

Limitations. The approach we have presented here has its limitations. The computation of the new vector of a word might exhibit inaccuracies. An example is the inaccuracy in the word movement through normalization, although the magnitude of this effect seems negligible. Another problem is that feedback at a new stage can overwrite the feedback of a previous stage. For example, the word *ociz* was moved in the second stage towards the word *ocis*, and in the third stage to the word *morz*, which results in overwriting the previous feedback. A similar problem emerges when a word is moved to two different words during the same feedback session. A solution for both cases would be to move a word to a position where it can satisfy both conditions, but such a position does not exist for many cases. Moving two words closer to each other in the same iteration leads to the same problem. For these cases, the current solution changes the vectors of these words before all of the other words to prevent indirect changes. Another limitation of our system is that the re-computation of the alignments is done stage-wise after multiple interactions, leading to feedback delays for a single interaction. Real time computation is currently not feasible because computing the WMD for two sentences is too complex. The resulting system was designed and developed in close collaboration with one expert reader (*DJW*) to address his needs. Consulting other users could guide us to other feedback visualizations fulfilling different information needs. Furthermore, the resulting model might be fitted towards the needs and interests of *DJW*, and this genre, one of the most complex, if not the most complex, in medieval French from the perspective of *mouvance*. Different expert readers could be interested in different relations between the words and lines, resulting in different interactions with the system and in different word vectors. Furthermore, the main features our system processes consist of sentences, words and character n-grams, although our focus lies in the alignment of epic poetry. For the comparison of poetry, focusing on syllables might further improve, achieving new results when comparing related texts.

Future Work. Our stage-based interdisciplinary exchange not only led to many ideas implemented in the course of the project but also identified room for future improvements. For example, visualizations could provide more granular information on how user feedback changed the vector space. The current solutions do not convey how the scoring of one particular alignment during an entire feedback session affected the whole word vector space. Moreover, a comparative view on the influence of different scorings, e.g., low vs. high scorings, could provide valuable information on how feedback is processed and on optimal use of the system. An active learning approach could support the scholar in generating feedback, for example, an algorithm based on string similarity and vector similarity could select pairs of likely synonyms and variants to be presented to and approved by the scholar. This could, on the one hand, ease the scoring session, and support the generation of dictionaries valuable to domain scholars on the other. A setup with multiple scholars refining a single vector space could also be of interest for its social, collaborative potential in the humanities. The visualization of similarities and differences in their interactions with the system could provide valuable information for future developments in visualization. Another interesting aspect would be focusing on a larger collection of vernacular literature, although the corpus of texts exhibiting such a high level of *mouvance* is somewhat limited. A visualization system for comparing more than two texts in the same traditions could direct the scholar to hitherto unknown alignment patterns.

7 Conclusion

Our interdisciplinary collaboration began several years back with the goal of establishing a system that supports the semi-automated alignment of unstable, medieval text versions. Our journey took us first to a parameter-based white-box approach (*iteal-V1*) to compute alignments on the basis of syntactic text features [3], and second to a black-box approach based on word embeddings (*iteal-V2*) that also considers semantic text features and thematically related concepts [5].

This work on *iteal-V3* documents our efforts developing a series of visualizations capable of conveying the complex structure of the word vector space intuitively for professional reading. Furthermore, we created means to integrate scholarly feedback back to the vector space model, and visual cues to observe how user-driven modifications affect local neighborhoods in the vector space. Our stage-based development process attests to the fact that explainable visualizations like the ones presented in this paper are capable of building bridges between computer science and other domains, thereby expediting gradual trust building in complex algorithmic processes. Benefiting from the expert reader's feedback, the word embedding approach finally led to a better performing alignment computation transferable to related text-based scenarios.

Our project was, and will continue to be, carried out as a participatory design process, from which all members gain valuable inspirations that can be carried out to the respective research areas. With our documented process we hope to inspire other visualization researchers to engage in participatory design, which pays off in the form of numerous ideas for future research directions.

References

[1] P. Zumthor, *Toward a medieval poetics*. U of Minnesota Press, 1992.

[2] T. Yousef and S. Jänicke, "A survey of text alignment visualization," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[3] S. Jänicke and D. J. Wrisley, "Interactive visual alignment of medieval text versions," in *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2017, pp. 127–138.

[4] R. Siemens, C. Leitch, A. Blake, K. Armstrong, and J. Willinsky, ""it may change my understanding of the field": Understanding reading tools for scholars and professional readers," vol. 3, no. 4, 2009.

[5] C. Meinecke, D. J. Wrisley, and S. Jänicke, "Automated Alignment of Medieval Text Versions based on Word Embeddings," in *LEVIA 2019: Leipzig Symposium on Visualization in Applications*, 2019.

[6] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[7] M. Sample, "The black box and speculative care," *Debates in the Digital Humanities*, pp. 445–448, 2019.

[8] J. Van Zundert, S. Antonijević, and T. Andrews, "6. 'black boxes' and true colour — a rhetoric of scholarly code," *Digital Technology and the Practices of Humanities Research*, pp. 123–162, 02 2020.

[9] S. Jänicke and D. J. Wrisley, "Visualizing mouvance: Toward a visual analysis of variant medieval text traditions," *Digital Scholarship in the Humanities*, vol. 32, no. suppl_2, pp. ii106–ii123, 2017.

[10] S. Jänicke, P. Kaur, P. Kuźmicki, and J. Schmidt, "Participatory Visualization Design as an Approach to Minimize the Gap between Research and Application," in *The Gap between Visualization Research and Visualization Software (VisGap)*. The Eurographics Association, 2020.

[11] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International Conference on Machine Learning*, 2015, pp. 957–966.

[12] A. Abdul-Rahman, G. Roe, M. Olsen, C. Gladstone, R. Whaling, N. Cronk, R. Morrissey, and M. Chen, "Constructive visual analytics for text similarity detection," in *Computer Graphics Forum*, vol. 36, no. 1, 2017, pp. 237–248.

[13] S. Jänicke, A. Geßner, and G. Scheuermann, "A distant reading visualization for variant graphs," *Proceedings of the Digital Humanities*, vol. 2015, 2015.

[14] S. Jänicke, A. Geßner, M. Büchler, and G. Scheuermann, "Visualizations for Text Re-use," in *Information Visualization Theory and Applications (IVAPP), 2014 International Conference on*. IEEE, 2014, pp. 59–70.

[15] P. Riehmann, M. Potthast, B. Stein, and B. Froehlich, "Visual assessment of alleged plagiarism cases," in *Computer Graphics Forum*, vol. 34, no. 3, 2015, pp. 61–70.

[16] T. Yousef, "Ugarit: Translation Alignment Visualization," in *LEVIA 2019: Leipzig Symposium on Visualization in Applications*, 2019.

[17] B. Asokarajan, R. Etemadpour, J. Abbas, S. Huskey, C. Weaver, N. Andrienko, and M. Sedlmair, "Visualization of latin textual variants using a pixel-based text analysis tool," in *Proceedings of the International EuroVis Workshop on Visual Analytics, EuroVA*, vol. 16, 2016.

[18] M. Behrisch, M. Krstajic, T. Schreck, and D. A. Keim, "The news auditor: Visual exploration of clusters of stories," in *Proc. EuroVA International Workshop on Visual Analytics*. Eurographics, 2012, pp. 61–65.

[19] S. Schreibman, "Versioning machine," "http://v-machine.org/", 2017, (Retrieved 2019-02-07).

[20] D. Wheeles and K. Jensen, "Juxta commons," in *In Proceedings of the Digital Humanities 2013*, 2013.

[21] M. Baumann, M. John, H. Pflüger, C. Herberichs, G. Viehhauser, W. Knopki, and T. Ertl, "An Interactive Visualization for the Analysis of Annotated Text Variance in the Legendary Der Heiligen Leben, Redaktion," in *LEVIA 2019: Leipzig Symposium on Visualization in Applications*, 2019.

[22] Z. Geng, T. Cheesman, R. S. Laramee, K. Flanagan, and S. Thiel, "Shakervis: Visual analysis of segment variation of german translations of shakespeare's othello," *Information Visualization*, vol. 14, no. 4, pp. 273–288, 2015.

[23] M. S. Alharbi, T. Cheesman, and R. S. Laramee, "Transvis: Integrated distant and close reading of othello translations," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[24] A. Hazem, B. Daille, D. Stutzmann, J. Currie, and C. Jacquin, "Towards automatic variant analysis of ancient devotional texts," in *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, 2019, pp. 240–249.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[26] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 746–751.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[28] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg, "Embedding projector: Interactive visualization and interpretation of embeddings," *arXiv preprint arXiv:1611.05469*, 2016.

[29] A. Kutuzov and E. Kuzmenko, "Building web-interfaces for vector semantic models with the webvectors toolkit," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 99–103.

[30] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 1992–2001, 2013.

[31] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu, "Interactive correction of mislabeled training data," in *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2019.

[32] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist, "Conceptvector: text visual analytics via interactive lexicon building using word embedding," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 361–370, 2017.

[33] N. Katricheva, A. Yaskevich, A. Lisitsina, T. Zhordaniya, A. Kutuzov, and E. Kuzmenko, "Vec2graph: a python library for visual-

izing word embeddings as graphs," in *International Conference on Analysis of Images, Social Networks and Texts*. Springer, 2019, pp. 190–198.
[34] A. Kashcha, "word2vec graph," "https://github.com/anvaka/word2vec-graph", 2018, (Retrieved 2020-02-27).
[35] M. El-Assady, R. Kehlbeck, C. Collins, D. Keim, and O. Deussen, "Semantic concept spaces: Guided topic model refinement using word-embedding projections," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 1001–1011, 2019.
[36] J. Choo and S. Liu, "Visual analytics for explainable deep learning," *IEEE computer graphics and applications*, vol. 38, no. 4, pp. 84–92, 2018.
[37] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi, "The state of the art in integrating machine learning into visual analytics," in *Computer Graphics Forum*, vol. 36, no. 8. Wiley Online Library, 2017, pp. 458–486.
[38] L. Jiang, S. Liu, and C. Chen, "Recent research advances on interactive machine learning," *Journal of Visualization*, vol. 22, no. 2, pp. 401–417, 2019.
[39] K. Kucher, C. Paradis, M. Sahlgren, and A. Kerren, "Active learning and visual analytics for stance classification with alva," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 7, no. 3, pp. 1–31, 2017.
[40] F. Heimerl, S. Koch, H. Bosch, and T. Ertl, "Visual classifier training for text document retrieval," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2839–2848, 2012.
[41] L. S. Snyder, Y.-S. Lin, M. Karimzadeh, D. Goldwasser, and D. S. Ebert, "Interactive learning for identifying relevant tweets to support real-time situational awareness," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 558–568, 2019.
[42] R. Makki, S. Brooks, and E. E. Milios, "Context-specific sentiment lexicon expansion via minimal user interaction," in *2014 International Conference on Information Visualization Theory and Applications (IVAPP)*. IEEE, 2014, pp. 178–186.
[43] T. Munzner, *Visualization analysis and design*. CRC press, 2014.
[44] K. Kallio, E. Mäkelä, and M. Janicki, "Historical oral poems and digital humanities: Starting with a finnish corpus," *Folklore Fellows' Network*, vol. 54, 2020.
[45] "Petit de julleville, louis." in *La Chanson de Roland : Traduction nouvelle rythmée et assonancée*. Alphonse Lemerre, Classiques français, 1878.
[46] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.
[47] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
[48] S. Tang, M. Mousavi, and V. R. de Sa, "An empirical study on post-processing methods for word embeddings," *arXiv preprint arXiv:1905.10971*, 2019.
[49] J. Mu, S. Bhat, and P. Viswanath, "All-but-the-top: Simple and effective postprocessing for word representations," *arXiv preprint arXiv:1702.01417*, 2017.
[50] K. Ethayarajh, "Unsupervised random walk sentence embeddings: A strong but simple baseline," in *Proceedings of The Third Workshop on Representation Learning for NLP*, 2018, pp. 91–100.
[51] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, 2019.
[52] T. Munzner, "A nested model for visualization design and validation," *IEEE transactions on visualization and computer graphics*, vol. 15, no. 6, pp. 921–928, 2009.
[53] D. J. Wrisley, "Pre-visualization," in *3rd Workshop on Visualization for the Digital Humanities, IEEE VIS 2018, Berlin, Germany*, 2018.
[54] A. Liu, "Toward a diversity stack: Digital humanities and diversity as technical problem," *PMLA*, vol. 135, no. 1, pp. 130–151, 2020.
[55] J. J. Lee, B. Greteman, J. Lee, and D. Eichmann, "Linked reading: Digital historicism and early modern discourses of race around shakespeare's othello," *Journal ISSN*, vol. 2371, p. 4549, 2018.
[56] W. von Wartburg, *Französisches etymologisches Wörterbuch*. F. Klopp, 1959.
[57] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and understanding neural models in nlp," *arXiv preprint arXiv:1506.01066*, 2015.
[58] J. Rocchio, "Relevance feedback in information retrieval," *The Smart retrieval system-experiments in automatic document processing*, pp. 313–323, 1971.
[59] S. Jänicke, "Valuable Research for Visualization and Digital Humanities: A Balancing Act," in *Workshop on Visualization for the Digital Humanities, IEEE VIS 2016, Baltimore, Maryland, USA*, 2016.
[60] U. Hinrichs, M. El-Assady, A. J. Bradley, S. Forlini, and C. Collins, "Risk the drift! stretching disciplinary boundaries through critical collaborations between the humanities and visualization," in *2nd Workshop on Visualization for the Digital Humanities, IEEE VIS 2017, Phoenix, Arizona, USA*, 2017.
[61] A. J. Bradley, M. El-Assady, K. Coles, E. Alexander, M. Chen, C. Collins, S. Jänicke, and D. J. Wrisley, "Visualization and the digital humanities:," *IEEE Computer Graphics and Applications*, vol. 38, no. 6, pp. 26–38, Nov 2018.
[62] V. Schetinger, K. Raminger, V. Filipov, N. Soursos, S. Zapke, and S. Miksch, "Bridging the gap between visual analytics and digital humanities: Beyond the data-users-tasks design triangle," in *4th Workshop on Visualization for the Digital Humanities, IEEE VIS 2019, Vancouver, British Columbia, Canada*, 2019.
[63] A. Abdul-Rahman, J. Lein, K. Coles, E. Maguire, M. Meyer, M. Wynne, C. R. Johnson, A. Trefethen, and M. Chen, "Rule-based visual mappings – with a case study on poetry visualization," *Computer Graphics Forum*, vol. 32, no. 3pt4, pp. 381–390, 2013.
[64] S. Jänicke, J. Focht, and G. Scheuermann, "Interactive visual profiling of musicians," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 200–209, 2016.
[65] P. Meyer, "Étude sur les manuscrits du roman d'alexandre," *Romania*, vol. 11, no. 42/43, pp. 213–332, 1882.
[66] T. Underwood, "A genealogy of distant reading." *DHQ: Digital Humanities Quarterly*, vol. 11, no. 2, 2017.
[67] K. Bode, "The equivalence of "close" and "distant" reading; or, toward a new object for data-rich literary history," *Modern Language Quarterly*, vol. 78, no. 1, pp. 77–106, 2017.
[68] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann, "On close and distant reading in digital humanities: A survey and future challenges." in *EuroVis (STARs)*, 2015, pp. 83–103.

Christofer Meinecke received his MSc degree in Computer Science from Leipzig University in 2019. Currently, he is a PhD candidate and research assistant at the Image and Signal Processing Group at Leipzig University, Germany. His research interests include text visualization, visualization in the humanities and machine learning for the humanities.

David Joseph Wrisley is Associate Professor of Digital Humanities at New York University Abu Dhabi, UAE. He holds a PhD in Literature from Princeton University, USA. His research interests include translation and variance in premodern textual traditions, spatial humanities as well as multilingualism and multidirectionality in digital humanities.

Stefan Jänicke is Associate Professor for Data Science at the Department of Mathematics and Computer Science at the University of Southern Denmark, Odense, Denmark. He received his PhD degree in computer science in 2016 from Leipzig University, Germany. His research interests relate to information visualization and visual analytics with a focus on text-, time- and geo-visualization, and their application in research areas like (digital) humanities, linguistics, social sciences, biology and sports.