AlignVis: Semi-automatic Alignment and Visualization of Parallel Translations

Mohammad Alharbi*, Tom Cheesman*, Robert S. Laramee*†
*Swansea University, Wales
{m.s.alharbi, t.cheesman, r.s.laramee}@swansea.ac.uk
†University of Nottingham, England
robert.laramee@nottingham.ac.uk

***Abstract*—Digital humanities and translation scholars utilize off-the-shelf tools to align multiple related translations. These tools generally rely solely on domain expert knowledge and do not exploit the recent advancements in computational linguistics and text mining. This paper presents AlignVis, a visual tool that provides a semi-automatic alignment framework to align multiple translations. It presents the results of using text similarity measurements and enables the user to create, verify, and edit alignments using a novel visual interface. The design consists of three main components: the alignment editor canvas, the post-edit area, and the user options panel. AlignVis exploits both close and distant reading and is designed to help digital humanities and translation scholars enhance the process of text alignment for multiple translations. The design of AlignVis is driven by iterative discussions with the domain expert which resulted in five benefits: presenting an overview of the aligned translations, support for multiple alignments, enhancement and acceleration of the alignment process, alignment refinement, and testing different similarity measurements. We evaluate AlignVis with domain expert feedback and a comparison with a standard alignment tool and computational and visual alignment tools.**

***Index Terms*—Information Visualization, Parallel Translations, Alignment**

I. INTRODUCTION

scholars may invest substantial time and effort studying the correspondence between a source text and its translations. The term "source text" refers to any text an expert would like to translate. The alignment between two stable translations or editions can sometimes be straightforward. A translation is considered "stable", if it does not vary much between other translations of the same source text. When there is a considerable amount of uncertainty, orthography reforms, or translation instability, manual alignment becomes a challenging, tedious, and error-prone task.

Furthermore, digital humanities and translation scholars spend considerable time using standard text alignment tools, such as LF-Aligner [1] to create parallel corpora of translations. Most of the tools require, and rely heavily on, domain experts to manually segment and validate translated texts one-by-one. We hypothesize that the recent advancements in text mining and computational linguistics can address these challenges.

In this paper, we present AlignVis, a tool that facilitates the advancement of text alignment techniques and provides interactive visual methods for the domain expert to edit and validate text alignments. AlignVis combines interactive visualization and domain knowledge intervention techniques in order to support exploration, validation, and refinement of machine recommended alignments. It enables the user to compare and test multiple text representations and similarity measurements to accelerate the alignment process.

The development of AlignVis is a product of collaborative work between computer scientists and a specialist in modern and contemporary German literature and culture. We test AlignVis with a collection of Othello translations into German which includes 38 translations as well as the English text of a sample of *The Tragedy of Othello, the Moor of Venice* play (1604) –Act 1 Scene 3.

Contributions: In this paper, we contribute the following:

- A novel visual alignment tool to help translation scholars align **multiple texts** simultaneously.
- An interactive visual interface to help **enhance and accelerate the alignment** process and enable modification of the alignments.
- Domain expert feedback, a comparison with a standard alignment's tool, and a comparison with visual and computational alignments' tools.

The rest of this paper is organized as follows: Section II discusses previous work related to our approach. Section III outlines the design requirements. Section IV explains the parallel translation data and the relevant terminologies. Section V introduces the design of AlignVis. Section VI is dedicated for the evaluation of AlignVis.

Please see the supplementary video for demonstration: https://youtu.be/binnROzTwOc

II. RELATED WORK

The related work is divided into four sections as follows. We searched for both surveys [2] and books [3] on these topics. We summarize the first section in Table I.

• **Visual Designs for Comparison of Parallel Texts:** Several approaches facilitate visual design to support comparative tasks. A juxapositioning, side-by-side layout is common to visualize and compare multiple documents ([12]–[16]).
Other approaches utilize the text's existing hierarchies and visualize the text properties using pixel-based visualization, such as Oelke and Kim [17] and Asokarajan et al. [18], [19].

Reference	Year	Number of Aligned Text simultaneously	Close reading	Distant reading	Source of text studied
Melamed [4]	1989	2	bipartite graph	-	The Bible translations
Cairo [5]	2000	2	juxtapositioned word-to-word corresponds	-	Italian-English corpus
Tufiş [6]	2006	2	bipartite graph	-	Romanian-English parallel corpus
Yawat [7]	2008	2	juxtapositioned word-to-word corresponds	dot plot matrix	German-English corpus
SWIFT Aligner [8]	2014	2	bipartite graph	-	French-English corpus
Jänicke et al. [9]	2014	2	bipartite graph, variant graph	heat-map, dot plot	The Bible translations
iTeal [10]	2017	Multiple	bipartite graph, variant graph	juxtapositioned alignment map	Literature
ViTA [11]	2017	2	bipartite graph	dot plot graph	Literature

TABLE I

A SUMMARY TABLE OF RELATED WORK AND PAPER CHARACTERISTICS INCLUDED IN THE COMPUTATIONAL AND VISUAL ALIGNMENT SECTION. THE DASHES (-) IN THE DISTANT READING COLUMN INDICATE THAT THE CORRESPONDING REFERENCE DOES NOT FEATURE DISTANT READING.

There are other interfaces that overlay parallel texts in the same coordinate system to facilitate comparison. For example, the variant graph [20], storylines [21], and parallel coordinates [14], [22] illustrate each compared object, such as a document, sentence or word, as a line in the visual space. In variant graphs, the y-axis depicts the offset in the text or time.

There are approaches that visualize relationships between parallel documents to support visual comparison. For example, the stylometric representation encodes similarity between versions using the thickness and length of the links [23]. Dot plots representation is also used to explicitly encode document relationships and to detect similarity and other patterns [9], [11], [24]. Collins et al. [25] also encode the relationships between documents using links and word clouds.

• **Computational and Visual Alignment:** Jänicke and Wrisley [10] and Abdul-Rahman et al. [11] integrate similarity measurements to detect alignment between texts. Jänicke and Wrisley introduce an interactive visual analytics tool (iTeal) that facilitates computational alignment between multiple text editions. They also provide imagery for different hierarchy levels (entire text, lines and words). Abdul-Rahman et al. [11] propose a web-based visual analytics tool (ViTA) that enables domain experts to interfere in the text alignment pipeline.

Further, translation studies scholars use different off-the-shelf tools [26]–[28] to segment and align parallel texts in practice. Most of the tools utilize a user interface which enables the user to choose the source and the target texts then present the alignment results in a tabular form. They also enable the user to perform certain functions to post edit the segments and alignments, such as splitting, merging or deletion.

Also, there are multiple approaches which offer visualizations of the pre-defined alignments and support for manual annotation. For example, Melamed [4], Ahrenberg et al. [29], Yawat [7], Tufiş [6], and SWIFT Aligner [8]. On the other hand, there are approaches, such as Cairo [5], which align corresponding words and do not allow the user to post-edit the alignments. Most of these approaches provide a word-based alignment and use a simple bipartite graph to visualize the links between the words.

LF-Aligner [1] is one of the standard tools that is commonly used to align translations. LF-Aligner supports multiple languages and parallel translations. We provide a comparison between LF-Aligner and AlignVis in Section VI-C. We also provide a comparison between our design and LF-Aligner, ViTA and iTeal in SectionVI-B.

AlignVis is different from these approaches since it combines a visual design that enables multiple alignments simultaneously and enables user's interference and modification of the result. AlignVis also enables the user to test the result based on different similarity metrics.

• **Text Re-use and Plagiarism Detection:** Multiple approaches have been developed to detect commonality between texts. Jankowska et al. [30] use n-grams to generate a relative n-grams signature to compare multiple texts against a base text. Jänicke et al. [9] introduce multiple visual designs such as heat-map display to depict text re-use patterns in English Bible translations. Jänicke et al. use a 2D dot plot to show the text re-use patterns between two texts. Abdul-Rahman et al. [11] use a 2D similarity metrics plot to facilitate the discovery of different text re-use patterns,. The Versioning Machine [12] and JuxtaCommons [31] are digital humanities tools which visualize corresponding text fragments using color highlights and links.

Similarly, visualization techniques have been used to facilitate detection of plagiarism between texts [24], [32]–[36].

Most of the work presented in text re-use and plagiarism present a visual design that does not incorporate user's knowledge in the alignment process and does not enable the user to test different similarity measurement methods. The purpose of these tools is mainly to indicate repeated text in a binary fashion.

• **Version Control Systems:** Although, software evolution visualization approaches [37], [38] differ from our work, they feature some overlap as they try to compare and visualize the similarity and differences between source code. McNabb and Laramee [2] include eight surveys in the software visualization category such as, Caserta and Zendra [39] and Mattila et al. [40]. Merino et al. [41] review 387 software visualization approaches. Novais et al. [42] attempt to provide structure for the evolution of visualization approaches field.

The approaches presented in this section serve a different purpose and do not support multiple alignments of texts. The goal of these tools is to show the edit history of source code text.

III. Requirement Analysis

Throughout our discussions with the domain expert, various tasks were identified. Humanities scholars, when studying divergent translations of literature, are interested in combining both distant and close reading. Domain experts also appreciate machine assistance to support and speed up the processes of comparative interpretation. The domain expert we collaborate with has previously tested different similarity measurement and would appreciate the ability to explore and observe the different results each measurement produces.

The requirements were derived and incrementally refined based on multiple meetings with the domain expert as explained in Section VI. We couple the requirements to the discussion of our design.

RO. Provide an overview of the aligned translations: The domain expert is interested in an overview of the aligned translations to analyse and explore the variation among them. The domain expert would like to explore the overall relations between translations.

RN. Support for multiple alignments: The domain expert is interested in a design that facilitates and integrates alignments for multiple translations.

RA. Enhance and accelerate the alignment process: Given that the current practice of alignment is time-consuming, error-prone, and performed one-by-one, translation scholars are interested in a tool that enables them to enhance and accelerate the process of alignment.

RE. Allow the user to refine and update the alignments: The result of the automatic alignment is not always accurate due usually to the instability and variation in translations. Therefore, the domain expert would like to be involved in the process of alignment and update the semi-automatic alignment process manually. This requirement is closely linked with the requirement RA. Yet, RE focuses on incorporating the expert user knowledge which results in enhancing and accelerating the alignment process.

RS. Enable the user to apply and test different similarity measurements: The domain expert would appreciate exploring different similarity measurements and examine the results that each measurement produces. They may observe if the results agree with the domain expert's own understanding.

IV. Description of Parallel Translation Data and Terminology

A group of researchers from the arts and humanities established a project called "Translation Arrays: Version Variation Visualization (VVV)" [43]. They collect parallel translations of Shakespeare's work and apply digital humanities and visualization techniques in order to explore and analyse the collection. The project website [44] hosts the parallel corpora. Each corpus consists of an English text and multiple translations. We test AlignVis with the collection: *The Tragedy of Othello, the Moor of Venice* play (1604) –Act 1 Scene 3. In this corpus, one text is in English and there are 38 manually aligned German translations. The translations were optically scanned from paper prints, corrected for OCR errors and segmented. They were collected over a time-span of 2-3 years from various sources, such as libraries, second-hand book-sellers, archives, theater publishers and theater companies. The translation data is stored in XML format on the project's website. In the following, we describe some domain-related terminology:

• **Segment [s]**: text that contains one or more words based on the user's tokenization preferences (usually a sentence).

• **Alignment [$a(s_i, s_j)$]**: consists of two segments, (s_i, s_j), that are related to each other. The machine-recommended alignment is the result of the pre-processing phase. We use the notion $a(T_1, T_2)$ to refer to an alignment between two translations T_1 and T_2. The notation $a(s_i, s_j)$ refers to an alignment between two individual segments s_i and s_j.

• **English Text [T_E]**: also called the source text. In our case, the source language is English, so we refer to to the source text as the English Text (T_E).

• **Focus Translation and Base Translation [T_F, T_B]**: can also be called the target texts. We feature two important target texts. The first is called the Base Translation (T_B) which the user chooses to represent the English text (T_E). The second is called the Focus Translation (T_F) which can be aligned with the T_B.

• **Sequential Alignments**: Sequential alignment is a common practice within the domain expert's practice. The alignment process is done by creating an alignment $a(s_i, s_j)$, where $s_i \in T_1$ and $s_j \in T_2$. Then a standard process creates the alignments $a(s_{i+1}, s_{j+1})$, $a(s_{i+2}, s_{j+2})$, etc. When there is a mismatch, the domain expert corrects it and start the process over. The process re-starts from the corrected mismatch.

• **Distance Value [$d(s_i, s_j)$]**: indicates the distance, d, between $s_i \in T_B$ and $s_j \in T_F$. The distance may vary based on the similarity measurements used.

• **Alignment Confidence Value[c]**: is a gradient operator. It measures the difference in distance values, $c = |d_1 - d_2|$, where $d_1 = d(s_i, s_j)$, and $d_2 = d(s_i, s_{j+1})$, and s_j, s_{j+1} are successive in the same translation. It is based on a heuristic used by the domain expert. If the distance between successive aligned segment pairs is high, this indicates a high certainty that the current segment alignment, $a(s_i, s_j)$, is correct.

A. Alignment Preprocessing

In order to derive similarity measurements and recommend matches between corresponding translation segments, we have a three-step preprocessing pipeline as shown in Figure 2. We first normalize the text, remove stopwords and sparse terms (optional), and tokenize the text **(1)**. Then, **(2)** we generate various embeddings that are used to compute similarity measurements. Embeddings include term TF-IDFs (Term Frequency–Inverse Document Frequency), term IDFs (Inverse Document Frequency) [45], [46], and contextual word embeddings (word2vec) [47]. After generating the embeddings, **(3)** we implement the similarity measurements to derive the matches between segments. For this, we use a selection of popular, state-of-the-art distance and similarity measurements. The first three are most often used with TF-IDF and IDF

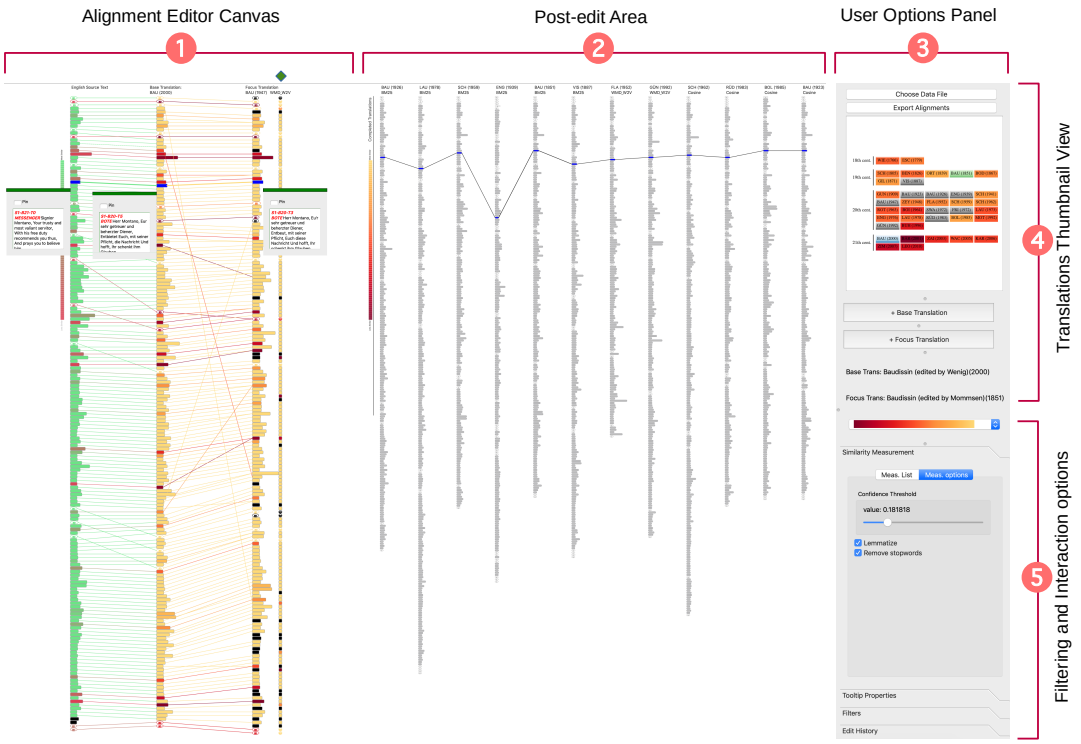


Fig. 1. An overview of AlignVis. The alignment editor canvas (1) illustrates the original English text (T_E), the base German translation (T_B), and the focus German translation (T_F). This view shows the machine-recommended alignments between the German translations and enables manual refinement. The column that is indicated by green diamond shows the secondary measurement feature,The post-edit area (2) shows the processed translations, the user can explore the content and move them back to the editor canvas. The user options panel (3) provides filtering and interaction options, also it includes the translation thumbnail overview (4). The latter view shows the translations in chronological order of publication and enables the user to add translations to the editor canvas. The user options panel provides options that enable the user to interact with the design and change properties, such as similarity measurements, filters, and color schemes.

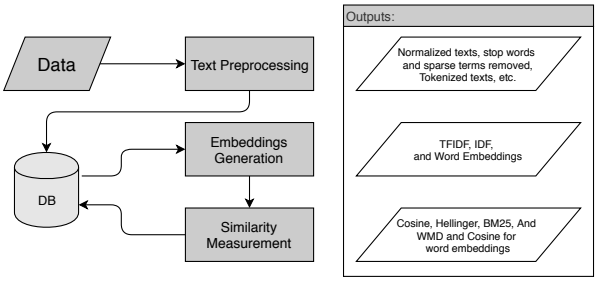


Fig. 2. The preprocessing pipeline illustrates three main stages. The text preprocessing process, where the text is normalized and the stop words can be removed, the embedding generation process, where the text is transformed into numerical vectors, and the similarity quantification, where the similarity measurement is preformed using a set of popular algorithms.

embeddings [48], and the last one is hypothesized to be the best that utilizes the quality of word2vec embeddings [49]. The similarity measurements we use are as follows:

Cosine Distance [d_{cosine}]: When two documents, T_1 and T_2, are represented as feature vectors, cosine distance is the angle between the vectors T_1 and T_2. The cosine distance is the dot product $T_1 \cdot T_2$ [50].

Hellinger Distance [$d_{hellinger}$]: Hellinger distance (or Bhattacharyya distance) is usually used to compute the similarity between two probability distributions. It can be used for both discrete and continuous distributions. In our case, since we are using bag-of-words features, distributions are discrete.

Okapi BM25: BM25 was developed as part of Okapi information retrieval system that was implemented at City University in London to retrieve a bibliographic reference database [51]. BM25 stands for "Best Matching" and is one of the variants of the BM best match function and is considered to be the most commonly used version [52].

Word Mover's Distance [d_{wmd}]: WMD is based on the results of the contextual word embeddings produced by word2vec [47]. Word embeddings are semantically meaningful representations for words generated using the local co-occurrences in a pre-defined window-sized neighborhood. The embeddings preserve the semantic relationships between words and enable arithmetic operators such as, vector('Berlin') - vector('Germany') + vector('France') $\approx$ vector('Paris') [47]. WMD calculates the distance between segments, d_{wmd}, and assumes that similar words have similar embeddings. WMD is designed to utilize word embedding relations and to overcome word transformations and reforms.

V. Design of AlignVis

In this section, we introduce the design of AlignVis and couple our choices with the requirements in Section III. Our tool utilizes automatic alignments exploiting a preprocessing phase discussed in Section IV-A. The design is composed of three main constituents, as shown in Figure 1. An editor canvas (1) that enables the user to view the machine-recommended alignments and refine them **(RA, RE)**, a post-edit area that provides an overview of the aligned translations (2) **(RO)**, and a user options panel (3) that enables the user to interact with the editor canvas and post-edit area **(RE)**. In the following sections, we discuss the design of AlignVis in more detail.

A. AlignVis Overview

In this section, we provide an overview of our tool's design components and how they address the requirements outlined previously.

Visual encodings in the alignment editor canvas: This view informs the current alignment process. The columns of rectangles, as shown in Figure 1 ❶, depict the English text (T_E), the base translation (T_B) and the focus translation (T_F) from left-to-right respectively. The rectangles present the text's segments top-down as it appears in the text. The length of each rectangle encodes the length of the text. The edges illustrate an alignment between two segments. The edges and rectangles are colored to show the confidence of the alignment and can be filtered based on the confidence value. In this view, the user is able to examine the machine-recommended alignments and refine them as necessary **(RA, RE)**.

Design Justification of the alignment editor canvas: AlignVis uses distant reading and juxtaposition with explicit encodings of the translations to facilitate the comparison between aligned sections [53] **(RA)**. Also, the process of alignment editing is a process involving close reading and it is more natural to the reader to read top-down. Juxtaposition and top-town order are consistent with previous tools. All of the interactions with the segments in this view are consistent and start with a right-click to accelerate the editing process and remain intuitive.

Visual encodings in the post-edit area: The second view is the post-edit area (Figure 1 ❷). This view stores the processed translations **(RO, RN)**. The most current translation is always placed on the left and while the remaining translations are shifted to the right. Similar to the alignment editor canvas the translations are illustrated using rectangles to depict the segments and, to use the space efficiently, the rectangles are rendered 40% smaller than the rectangles in the alignment editor canvas with respect to both the width and height. This view is linked with the alignment editor canvas, when the user highlights (using on-mouse-over) a segment, the aligned segments in the T_F and the aligned translations are also highlighted and a tooltip with the underlying text is shown **(RO, RN)**.

Design Justification of the post-edit area: AlignVis presents the processed translations and links them with the alignment editor canvas to help and guide the user throughout the alignment process. This is a key novel feature that enables the user to align multiple translations **(RN)**. They are ordered from left to right, with the most recent on the left. This makes it easier for the user to keep track of the processed translations. The post-edit view presents a distant reading of the processed translations which can guide the domain expert to similar translations while aligning the T_F.

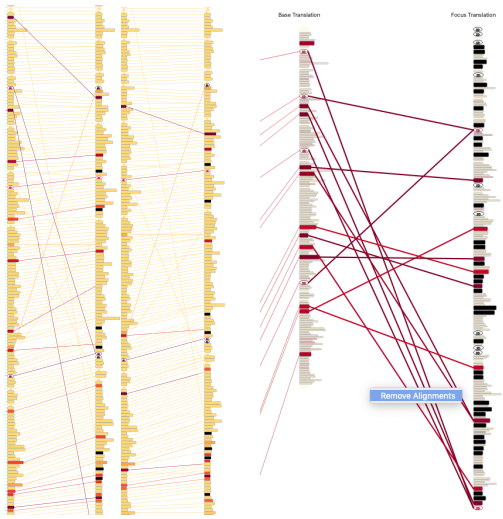
The third component is the user options panel (Figure 1 ❸). In this panel, we provide a thumbnail view of all translations (Figure 1 ❹) **(RO)**. The user options panel incorporates interaction and exploration means to customize and update the editor canvas and post-edit area.

Design Justification of the translation thumbnail view: This view presents the translations in chronological order to facilitate the search of a translation **(RO)**. The translation thumbnails are colored based on the average confidence value of each alignment. The confidence value indicates how certain the similarity measurement **(RS)** is after segment comparison. This is explained further in Section V-C. This color choice directs the user's attention to the level of alignment's certainty for each translation **(RA)**.

In the options panel, we provide various filtering and interaction options (Figure 1 ❺) that facilitate alignment and exploration **(RA)**. The options are organized and grouped based on their objectives. For example, all of the options that are related to changing and updating the similarity metrics are placed in one group called "similarity metrics". Also, all of the options that filter the data or design items are in the "Filters" tab. From the options panel, the user also can export the alignments into XML format that comply with the VVV's project format. The options panel also provides the user with multiple preset color schemes [54]. Th options presented in this panel are guided by our discussion and feedback with the domain expert.

B. Workflow Overview

In this section, we explain how the user interacts with AlignVis. The translation thumbnail view serves as a starting point, presents the collection, and illustrates the global similarity between translations. This is based on the average aggregation of the confidence values for each T_F. When the user selects a T_B, he can align it with the T_E sequentially. This alignment is to establish the link between the T_F and T_E. Afterwards, the alignment is confirmed by the user. The translation thumbnail view color scheme indicates the alignment certainty of each translation. This feature guides the user when selecting the T_F. Then, the alignment editor canvas renders an overview of the alignments between the T_B and T_F. AlignVis enables the user to change the similarity measurements to explore multiple matching segment results **(RS)**. It also enables the user to update the alignment based on a confidence value threshold, κ. If κ is below the threshold, AlignVis chooses the segment with the edge distance to the T_B from the best three matching segment candidates. AlignVis also facilitates interaction methods that can be used to help in the alignment exploration and verification such as the two-range filter of the



(a) (b)

Fig. 3. (a) On the left, the alignments without applying the confidence value threshold. On the right, the effect of applying a threshold (κ = 0.75). Some of the diagonal edges are removed. (b) A screenshot of the deletion action when the user selects multiple alignment edges.

confidence values and the secondary measurement to explore other measurements without forgetting the current one. The user may validate the machine-recommended alignment with previously processed alignments from the post-edit area.
AlignVis enables the user to modify and update the machine-recommended alignment with simple interactive actions. Each feature is easy to find when the user right-clicks on a segment, such as addition, update, or deletion. After aligning T_F, the user adds this T_F to the post-edit area and repeats the process of alignment with another T_F. The post-edit area is integrated with the alignment editor canvas and the user can move the translation between the two views interactively.

C. Semi-automatic Alignment Exploration and Verification

The exploration and verification of the alignments contribute to **(RA)** and **(RE)** and are particularly important since the user of AlignVis does not necessarily have experience with what AlignVis offers and the text similarity measurements.

AlignVis color maps the edges between the T_B and T_F based on confidence values. The user can verify the performance of the similarity measurement by the overall view in the alignment editor canvas. AlignVis incorporates a feature that allows the user to set a confidence value threshold, κ. If the alignment confidence value is below κ, AlignVis chooses the index shortest distance to the T_B segment from the best three segment candidates produced by the similarity measurement algorithm. The index distance is the difference between the index of the T_B's segment and the index of the T_F's segment. Figure 3a illustrates the effect of applying the confidence value threshold κ. In the alignments on the left, we see there are some diagonal edges which are probably not correct. On the other hand, the alignments on the right illustrate that these edges are reduced when applying a threshold of κ = 0.75.

Exploration and Verification in the Alignment Editor Canvas: The alignment editor canvas also implements an action that facilitates the reading of the T_F as well as verifying the T_B alignments **(RA)**. This is performed by selecting any segment, s of the T_F or T_B and then using the keyboard arrows to navigate to the next or previous segment. The segments and edges are highlighted while the user close-reads the translation as shown in Figure 1.

The user also can choose to add secondary similarity measurements to compare with the current measurement (Figure 1 ◆). This feature can help the user discover the best similarity measurement alignment if the first measurement fails **(RS)**. This feature was added after exploring the variance between the similarity measurements. A secondary measurement could recommend and improve the alignment and accelerate the alignment process **(RA, RS)**.

Exploration and Verification in the Post-edit Area: The post-edit area can be used to verify the correctness of the machine-recommended alignment **(RN)**. When the user selects a segment in the T_F or T_B, the post-edit area highlights the processed alignments and displays the original text if the user chooses. Also, an edge is rendered between segments $s_1 \rightarrow s_2$ to show $a(s_1, s_2)$ and to help capture a sense of the segment placements in the processed translations. For example, if the segment is aligned with two segments and the post-edit context view does not show this split alignment, this may indicate an incorrect alignment.

Sequential Alignments: There are cases where the alignment between two translations is even difficult for the domain scholars. Some of the translations are not stable and may be unfaithful with respect to the original T_E. They do not expect the machine-recommended alignments to detect many correct alignments. Therefore, we support the domain expert to perform sequential alignments for such cases. See Section IV for detailed explanation of sequential alignment.

In AlignVis, we implement this for aligning both the T_E with the T_B and T_B with the T_F. This follows the same convention, the user right-clicks on a T_B segment and from the menu and selects "Create Sequential Alignments". AlignVis follows the same process described previously and applies a domain constraint to match the segments types when aligning. The constraint enables AlignVis to only align a speech segment with a speech segment and a stage direction segment with a stage direction segment.

D. Domain Expert Refinement

AlignVis enables the domain expert to refine and update the machine-recommended alignments **(RN)**. There is a selection of editing tasks that AlignVis can offer.

Alignment addition: AlignVis enables the user to add a new alignment edge. The user can select a T_B segment, then right-click and choose "begin manual alignment". Then the alignment editor canvas changes to edit mode. When the user chooses any segment in the T_F, a dynamic edge is rendered and both segments are highlighted to facilitate the alignment process. The user can confirm the alignment by right-clicking

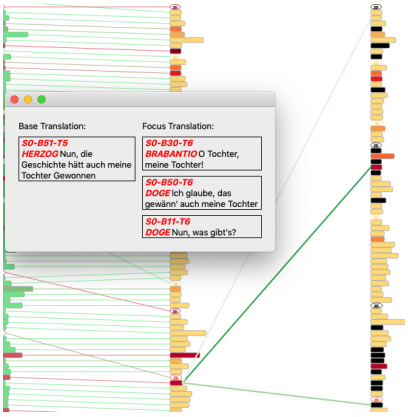


Fig. 4. A screenshot of the top three candidate segments for a user-chosen T_B segment **(RE)**. The candidate segment edges are rendered in green and the saturation of the color represents the rank of the candidate segments. The order of the segments is based on the ranking of the each segment.

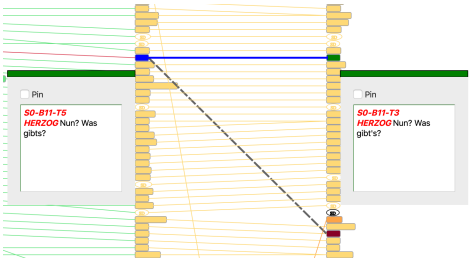


Fig. 5. The alignment update mode: when the user chooses to update an existing alignment **(RE)**. The original alignment is represented using a dashed gray edge, and a dynamic blue guide edge is rendered to indicate the new alignment.

and then choosing "confirm alignment". The user can also exit edit mode by choosing "clear alignment" or hitting the "ESC" key.

Furthermore, to accelerate the alignment process, the user can view the top three candidate segments based on the current similarity measurement by right-clicking on the base segment and choosing "Show candidates for alignment". Then, as shown in Figure 4, a separate window is presented to close-read the best three candidates for the segment ranked by distance. The saturation of the alignment edges represents the rank of each individual segment distance. After the user examines the original, he re-aligns the segment with the best match. The best three candidates for the segment are the first three focus segments with the shortest distance to the base segment.

Alignment update: The user can update an existing alignment and change both the T_F or T_B segment correspondence. A dynamic blue guide edge is rendered to indicate the new alignment. In order to help the user update the connections and not loose track of the original segment, the original segment is highlighted using a dashed gray edge as shown in Figure 5.

Alignment deletion: AlignVis incorporates the action of alignment deletion. Consistent with the previous actions, the user may right-click on an edge and choose "Delete an alignment edge". AlignVis also enables the deletion of multiple connections. The user can select multiple alignments then select "Delete alignment edges" to delete the selected alignments as shown in Figure 3b. All of the editing actions are stored and can be undone by the user.

E. Selection and Filtering

AlignVis incorporates a number of interaction features to customize the alignment editor canvas and the post-edit area **(RA)**.

Selection in the Alignment Editor Canvas: AlignVis enables the user to interact with the visual design in different ways. When the user chooses a segment, all of the aligned segments in the post-edit area are highlighted and linked with edges while the other segments are rendered as context as shown in Figure **??**. Customized tooltips are also designed to provide close reading of the segments. The user can slide them anywhere in the editor and pin them in a specified location (analogous to a post-it note) for further exploration as shown in Figure 1.

Selection in the Translation Thumbnail View: The translation thumbnail view in the user options panel (Figure 1 ❹) enables the user to add T_Fs to the alignment editor canvas **(RN)**. The user can select a translation by right-clicking and choosing "add focus translation" or double-clicking on the translation thumbnail. This results in interactively sliding the current T_F to the post-edit area and replacing it with the new user-chosen T_F. The T_F also can be removed from both the alignment editor canvas and the post-edit area. The user also can change the T_B translation by selecting the translation thumbnail and clicking on "+ Base Translation" to add the T_B translation to the alignment editor canvas. When the user selects a new T_B the current T_B is replaced.

Filtering of Translation Segments: The user also can apply filtering options to reduce the visual complexity **(RA)** caused by many segments. AlignVis offers two filters, the first excludes stage directions. A stage direction is a sentence which instructs the director and actors of the play. Researchers are sometimes not interested in studying stage directions and removing them can reduce clutter from the scene. The stage directions are clearly distinguished from normal speeches, as can be seen in Figure 1 ❶. They are illustrated by ellipses and colored borders.

The second filter renders the alignment edges based on the confidence value. This option offers a range slider to enable the user to set the confidence value threshold κ. The filters option provides the user with dynamic feedback on the number of preserved and filtered segments.

When applying this filter, the alignment editor canvas highlights the segments and edges withing the filter's range. Edges and segments which are not within the range are not rendered in focus as shown in Figure 3b.

VI. Evaluation

AlignVis is designed in close collaboration with the domain expert to design a solution that addresses the requirements

outlined in Section III. In the following sections, we describe the domain expert feedback, a comparison with a standard alignment tool and computational and visual alignment tools.

Domain Expert Feedback Sessions: We held 13 sessions to develop our work. Each session, on average, lasted an hour. The total time duration of the sessions is 12 hours and 15 minutes. All of the sessions are video-recorded for post analysis. Our semi-structured interview questions were previously planned and guided by Hogan et al [55]. The first few sessions were mostly software demonstrations and specification adjustment by the domain expert. We incrementally adjust our design and implement features based on the domain expert interviews. These sessions evolved into hands-on use of the tool. Throughout the sessions, there were several patterns that we observed, such as, the discovery of design bugs and new domain-related information.

Domain Expertise: The domain expert is Professor Tom Cheesman, in the Department of Modern Languages, Translation and Interpreting in the College of Arts and Humanities, Swansea University. He is the principal investigator on the "Version Variation Visualisation" project. The project is responsible for collecting, aligning, and warehousing the dataset that we are examining, and other 'multi-retranslation' datasets. He has been researching German culture and translating German literature since the early 1980s. Professor Cheesman has been investigating the history of German translations of Shakespeare's play Othello since 2009, using both traditional qualitative methods (contextualised close reading) and experimental, quantitative, digital methods. Relevant online outputs, presentations, and published articles by him and his collaborators are listed on the project's website [44]. The articles include publications in *Digital Scholarship in the Humanities* [23] and *Journal of Data Mining and Digital Humanities* [56].

A. Domain Expert Feedback

In this section, we report some of the domain expert feedback on AlignVis features.

Alignment Exploration and Verification Feedback: The domain expert finds the features that AlignVis provides to enable the expert to explore and read helpful as he states that, *"The process of establishing and checking alignment is a process of reading the text, and AlignVis could help with that a good deal."* Further, the domain expert states that *"The alignment editor canvas provides a quick way to read and check the alignments."* Exploration and reading is facilitated using the alignment editor canvas and the post-edit area which integrates close and distant reading in the same view. *"I like the way that looks right away,"* the domain expert stated as he was experimenting with the two designs.

The domain expert uses the edge color as an indicator to validate the alignment. In this way the user can explore and validate the alignment by the overall view of the alignment results **(RO)**. The domain expert agrees that coloring the edges to indicate the similarity measurement certainty is a good idea. The domain expert also agrees that the overall view illustrates how much a manual alignment is needed for a specific T_F.

The domain expert emphasizes that presenting the top suggested segments saves time **(RA)** as he needs to read the translation to find the other candidates. Also, automatically adjusting the alignments to choose the shortest edges using the confidence value threshold is helpful and saves time exploring the alignments **(RA)**. *"This feature sensibly prioritises the top few best candidates rather than presenting all possible candidates, speeding up my decision-making,"* the domain expert states.

The secondary measurement is a good way to see multiple measurements in the same view **(RN)**. For example, the domain expert investigates the alignment between the segment *"Saal im herzoglichen Palast"* in the T_B Baudissin (2000) and the segment *"Ein Beratungszimmer"* in the T_F Baudissin (1962). d_{cosine} failed at detecting this alignment, however, the d_{wmd} detected the correct one because it utilizes the semantic word embeddings. Both of the words "*Saal*" and "*Zimmer*" have similar meaning that indicates a room.

The Post-edit Area Feedback: One of the most important advantages of the post-edit is that it presents distant reading of the processed translations. The domain expert thinks that distant reading in both the alignment editor canvas and the post-edit is beneficial as it helps in the comparison task between translations at a global level **(RN)**. Highlighting the corresponding alignment across the post-edit area is also useful as the user explores and validates the alignments. *"Doing the alignment process is not just preparation, however, as you do the alignment you discover things about the texts that you want to investigate more. This view facilitates this as I am interested in the other texts and I want to be able to retrieve them,"* the domain expert states **(RA)**.

Domain Expert Refinement Feedback: The domain expert finds it easy to refine the machine-recommended alignments. He appreciates that he can perform a one-to-many and many-to-one alignment for the first time. He also appreciates that all of the alignments actions are in one place when he right-clicks on a segment. *"From a design point of view it is very good compared to other alignment tools that I have had to deal with, this is quick, painless and easy to do"*, the domain expert states **(RA)**.

The domain expert also likes the update function when AlignVis particularly encodes the original alignment whilst choosing another alignment (as in Figure 5). The dashed gray edge that represents the original alignment has the advantage that the user can see what he is going to change.

Selection and Filtering Feedback: The stage direction filter can be used to reduce the design complexity. This accelerates the work as it saves time reading the speeches without the stage direction interrupting the reading and alignment refinement process **(RA)**. The two-range confidence value also is beneficial as it reduces the edges between the T_B and T_F especially when the T_F is not stable.

The domain expert states that, *"The option to filter alignments by level of confidence is useful in cases where the user has found that the low-confidence suggestions are of no use (they are all false), so user can save time and attention*

Tasks supported	LF-Aligner [1]	ViTA [11]	iTeal [10]	AlignVis
Close reading	×	×	×	×
Distant reading		×	×	×
Multiple alignment (n>2)	×		×	×
Post-edit interaction	×		×	×
Incorporating similarity measurement		×	×	×
Testing different similarity measurement				×

TABLE II
A COMPARISON BETWEEN ALIGNVIS AND THE RELATED WORK.

by excluding/deleting them. On the other hand there can be cases where low-confidence suggestions are worth examining specifically, e.g where the overall confidence level is low." **(RE)**.

The domain expert thinks that the selection of the translations is intuitive as the translation thumbnail view presents the translations in chronological order of publication. It is consistent with the other views using the right-click to add and remove a translation **(RA)**.

Moveable and Pinable Tooltips: The domain expert admires the design of the tooltip as it helps him pin the tooltip and move it around which facilitates the comparison tasks with other segments. The domain expert uses this a lot as he reads and explores the translations **(RA, RN)**. *"'Excerpting' is a fundamental technique in humanities research – meaning, we read something, and select a quote/excerpt from it, or paraphrase the selection/excerpt in our own words, and we make a note including the excerpt and a reference. You can call this 'manual text mining'. In exploring and comparing translations, it's very helpful to be able to do this inside the application,"* the domain expert states.

B. Comparison With the Computational and Visual Alignment Tools

Table II provides a comparison summary of the tasks and related work. We only list related work that visually or computationally generates alignments. We base our comparison on six supported tasks that we derived from the requirement analysis discussed in Section III. The first two tasks are (T1) close and (T2) distant reading. Close reading involves the process of carefully reading word-for-word and interpreting a passage to develop a deep understanding of the ideas contained in the text [57]. Humanity scholars appreciate access to the raw text [58] and this increases the trust in the implemented approach [59]. Distant reading, on the other hand, illustrates the global features of the texts using computationally and analytically abstracted visualization [59].

Most of the tools in Table II implement close reading solutions. However, LF-Aligner does not provide any distant reading of the aligned texts.

Humanities scholars spend a great deal of their time aligning multiple versions or translations. Most of the tools present only one-to-one alignment solution. In Table II, we see that most of the compared related work incorporates (T3) alignment of multiple texts. However, ViTA is limited to only two texts.

Post-editing of the results is also a feature that humanity scholars value. The scholars' knowledge intervention is always an important add-on when preparing and aligning texts. most of the compared related work provides interaction to support (T4) post-editing of the results with the exception of ViTA which does not support human post-editing.

LF-Aligner uses different measures to sequentially align texts and does not (T5) incorporate similarity measurements. The other related work computationally aligns multiple texts using similarity measurement algorithms. AlignVis enables the user to (T6) test multiple results from varying similarity measurements and visualize them to support exploration and analysis. The other related work does not integrate the ability to use different similarity measurements beyond the implemented algorithm.

C. Comparison With a Standard Alignment Tool

Working with parallel versions or translations, the task of alignment of segments – one-to-one, one-to-many, or one-to-nil alignments, in each direction – is a far more complex operation. Traditionally it is performed manually by the scholar. Some view it as a menial task which should be outsourced if possible. For most scholars, the process of performing alignment is an opportunity to gain new knowledge and understanding of how texts relate to one another. In this case, a tool is needed which supports the process and allows the user to intervene.

LF-Aligner: We offer a comparison with LF-Aligner because this is what the domain expert uses in our case. Also, the domain expert has tried other tools and found LF-Aligner is the most useful for the purpose of aligning related texts. However, the limitations of LF-Aligner and other tools that humanity scholars commonly use inspired this project.

In LF-Aligner, the user uploads a corpus. The software then performs an initial, automatic sentence segmentation and an alignment, using an algorithm which primarily inspects sentence lengths in sequence. Success is varied. Particularly in dealing with our German Shakespeare corpus, LF-Aligner's results are very unreliable due to the huge differences between sentence length sequences in different versions, and structural differences.

LF-Aligner's manual alignment correction interface is a tabular display which fills the screen: parallel full texts, displayed as columns; segments are displayed as rows. The software's initial segmentation and alignment can be modified by the user, manually, using a small set of keyboard controls to split or merge, insert or delete cells.

The simplicity of LF-Alinger's interface is a benefit to most humanities scholars. Nothing distracts from the view of the texts, which are the scholar's main focus of interest. LF-Aligner is suited to close reading. The user combines the segmentation and alignment process with detailed inspection of the texts. The alignment correction process is slow, but it brings the scholar new information: new knowledge which contributes to the interpretation of the texts.

One disadvantage of LF-Aligner is that it cannot cope with transposition: cases where a segment sequence $\{s_1, s_2\} \in T_1$

aligns with a sequence $\{s_2, s_1\} \in T_2$. The user must manually reorder the sequence in one of the texts. But this creates an inaccurate representation of the original text – a loss of significant information.

The main disadvantage of the LF-Aligner interface is that it offers no distant reading. The segmented and aligned, manually edited corpus can easily be exported into other systems, which do offer distant reading. But the segmentation and alignment process will be more efficient if the user can shift back and forth between close, full text view and a distant overview of text structures and alignment patterns. In the LF-Aligner interface, depending on screen settings, the user sees only the equivalent of one to two printed pages on the screen. Therefore it is impossible to obtain an answer to questions requiring an overview such as: Which passages exhibit a continuous one-to-one alignment? Which passages have no alignment or multiple alignments? Which passages align differently in different versions? This sets limitations on the amount of information the user can gain from the alignment process.

AlignVis is very much superior to LF-Aligner with respect to distant viewing in both the alignment editor canvas and the post-edit area. The default view of a text is a distant view, representing a sequence of segments as a narrow column of blocks. Visual features of the blocks represent segment types (e.g. speech text or stage direction) and computed features of the segment (length, etc) and its alignment(s). This system of representation enables the equivalent of multiple printed pages to be represented on a single screen, affording a rapid overview of corpus characteristics of interest to the researcher: lengths, segmentation structures, patterns of alignment, and passages of interest for editing purposes.

The focus of interest here is not the automation but the display and the manual correction options. The automated results are far better than the results obtained by LF-Aligner, leading to significant time saving in manual correction.

LF-Aligner's auto-alignment implements the hypothesis that a sequence $\{s_1, s_2, s_3, s_4\} \in T_1$ will normally align with $\{s_1, s_2, s_3, s_4\} \in T_2$. This factor could have more influence on AlignVis's metrics. The steep diagonals alignments are certainly false alignments. But the ability to accommodate transposed alignments is an advantage.

A crucial issue for a humanities user is not so much the performance of automated alignment but rather the ease of inspecting and correcting alignments. Here AlignVis has several major advantages over LF-Align. These include: distant viewing of the full corpus in the options panel thumbnails, with color coding indicating confidence levels, guiding the setting of editing priorities; distant viewing of texts, segments, and alignments for base text and focus text; distant viewing of aligned texts in the post-edit area, with easy switching of the T_F; visual representation of confidence values for alignments, to guide editing priorities; ease of obtaining a close view of segment text, for exact reading; ease of manual correction.

Users can rapidly scroll through T_B and T_F simultaneously, speed-reading successive segments while visually checking the defined alignments. This resembles the user experience in LF-Aligner, except that by default, only one pair of aligned segments is in view, alignments are represented by an edge rather than as contents of a row; and the edge can be selected for rapid editing. The visual encoding of confidence values enables the user to scroll very rapidly where confidence is high, whereas in LF-Aligner, each aligned pair must be visually checked.

It is easy to learn to use the AlignVis menus for editing alignments. Keyboard shortcuts could be suggested for frequently used commands. The AlignVis user focuses on one segment pair at a time: compared with LF-Aligner, this speeds up the process in 'cruise mode' where many successive alignments are one-to-one. The user can very easily 'skip' down or back up columns. Where alignment problems occur, the option to 'pin' segments of interest is very helpful for close reading, which often requires close comparison of segments across different passages of a text. This helps ensure that the alignment task is a knowledge-gaining process for the scholar.

Overall, AlignVis is a very promising design, combining text mining and language processing affordances with a practical solution to supporting the labor-intensive task of exact segment alignment. It makes this task much more efficient than existing tools. At the same time, it supports the potential for alignment checking and correction to be an integral part of the scholarly process of understanding and interpreting texts.

VII. Conclusion

In this paper, we present AlignVis. A tool that combines interactive visualization with domain knowledge intervention to facilitate the alignment of parallel translations. AlignVis is designed with close collaboration with the domain expert to implement five requirements (Section III). AlignVis was evaluated by domain expert feedback and a comparison with a standard alignment tool that is widely used and the computational and visual alignment tools. Please see the supplementary video for demonstration: https://youtu.be/binnROzTwOc

Future work includes more enhancement in the language preprocessing to improve the results. The domain expert also suggests integrating the similarity measurements to vote for the correct alignment. He also advises highlighting alignment patterns such as one-to-many and one-to-nil. Another future direction is to implement means to understand the alignment results and the reasoning behind their choices.

As a limitation, scalability is still considered a challenge when dealing with textual datasets. The ability to use a small screen and visualize multiple documents and provide close and distant reading is a challenging task that could be addressed in future research.

References

[1] A. Farkas, "LF aligner," http://sourceforge.net/projects/aligner/, accessed on 29.5.2019.

[2] L. McNabb and R. S. Laramee, "Survey of surveys (sos) - mapping the landscape of survey papers in information visualization," *Computer Graphics Forum*, vol. 36, no. 3, pp. 589–617, 2017.

[3] D. Rees and R. S. Laramee, "A survey of information visualization books," *Computer Graphics Forum*, vol. 38, no. 1, pp. 610–646, 2019.

[4] I. D. Melamed, "Manual annotation of translational equivalence: The blinker project," *Technical Report 98-07, Institute for Research in Cognitive Science*, 1998.

[5] N. A. Smith and M. E. Jahr, "Cairo: An alignment visualization tool," in *The International Conference on Language Resources and Evaluation*, 2000, pp. 552–554.

[6] D. Tufiş, "From word alignment to word senses, via multilingual wordnets," *The Computer Science Journal of Moldova (CSJM)*, vol. 14, no. 1, pp. 3–33, 2006.

[7] U. Germann, "Yawat :yet another word alignment tool," in *The ACL-08: HLT Demo Session*, 2008, pp. 20–23.

[8] T. Gilmanov, O. Scrivner, and S. Kübler, "Swift aligner, a multifunctional tool for parallel corpora: Visualization, word alignment, and (morpho)-syntactic cross-language transfer," in *LREC*, 2014, pp. 2913–2919.

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

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

[11] 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," *Computer Graphics Forum*, vol. 36, no. 1, pp. 237–248, 2017.

[12] S. Schreibman, A. Kumar, and J. McDonald, "The versioning machine," *Literary and Linguistic Computing*, vol. 18, pp. 101–107, 2003.

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

[14] 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. [Online]. Available: https://doi.org/10.1177/1473871613495845

[15] S. Howell, M. Kelleher, A. Teehan, and J. Keating, "A Digital Humanities Approach to Narrative Voice in The Secret Scripture: Proposing a New Research Method," *Digital Humanities Quarterly*, vol. 8, no. 2, 2014.

[16] S. Jänicke and D. Joseph 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, 09 2017. [Online]. Available: https://doi.org/10.1093/llc/fqx033

[17] D. A. Keim and D. Oelke, "Literature fingerprinting: A new method for visual literary analysis," in *2007 IEEE Symposium on Visual Analytics Science and Technology*, Oct 2007, pp. 115–122.

[18] B. Asokarajan, R. Etemadpour, J. Abbas, S. Huskey, and C. Weaver, "Visualization of Latin Textual Variants using a Pixel-Based Text Analysis Tool," in *EuroVis Workshop on Visual Analytics (EuroVA)*, N. Andrienko and M. Sedlmair, Eds. The Eurographics Association, 2016.

[19] ——, "TexTile: A Pixel-Based Focus+Context Tool For Analyzing Variants Across Multiple Text Scales," in *EuroVis 2017 - Short Papers*, B. Kozlikova, T. Schreck, and T. Wischgoll, Eds. The Eurographics Association, 2017.

[20] S. Jänicke, A. Geßner, G. Franzini, M. Terras, S. Mahony, and G. Scheuermann, "TRAViz: A Visualization for Variant Graphs," *Digital Scholarship in the Humanities*, vol. 30, no. suppl˙1, pp. i83–i99, 10 2015. [Online]. Available: https://doi.org/10.1093/llc/fqv049

[21] S. Silvia, R. Etemadpour, J. Abbas, S. Huskey, and C. Weaver, "Visualizing variation in classical text with force directed storylines," in *Workshop on Visualization for the Digital Humanities*, 2016.

[22] Z. Geng, R. S. Laramee, T. Cheesman, A. Ehrmann, and D. M. Berry, "Visualizing translation variation: Shakespearee's Othello," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, B. Benes, K. Moreland, C. Borst, S. DiVerdi, C. Yi-Jen, and J. Ming, Eds., 2011, pp. 653–663.

[23] T. Cheesman, K. Flanagan, S. Thiel, J. Rybicki, R. S. Laramee, J. Hope, and A. Roos, "Multi-retranslation corpora: Visibility, variation, value, and virtue," *Digital Scholarship in the Humanities*, vol. 32, no. 4, pp. 739–760, 2017.

[24] R. L. Ribler and M. Abrams, "Using visualization to detect plagiarism in computer science classes," in *The IEEE Symposium on Information Vizualization*, 2000, pp. 173–178.

[25] C. Collins, F. B. Viegas, and M. Wattenberg, "Parallel tag clouds to explore and analyze faceted text corpora," in *2009 IEEE Symposium on Visual Analytics Science and Technology*, Oct 2009, pp. 91–98.

[26] D. Briel, "Bligner," http://bligner.aligner.free.fr/, accessed on 07.06.2019.

[27] M. L. Forcada and R. Martin, "bitext2tmx: Bitext aligner/converter," http://bitext2tmx.sourceforge.net/, accessed on 07.06.2019.

[28] LinguisTech, "Sdl trados winalign," https://linguistech.ca/SDLTrados_WinAlign_E_TUTCERTT_I_PartI, accessed on 07.06.2019.

[29] L. Ahrenberg, M. Merkel, and M. Petterstedt, "Interactive word alignment for language engineering," in *European Chapter of the Association for Computational Linguistics*, 2003, pp. 49–52.

[30] M. Jankowska, V. Kešelj, and E. Milios, "Relative n-gram signatures: Document visualization at the level of character n-grams," in *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2012, pp. 103–112.

[31] D. Wheeles and K. Jensen, "Juxta commons," *Proceedings of the Digital Humanities*, vol. 5, p. 12, 2013.

[32] D. R. White and M. S. Joy, "Sentence-based natural language plagiarism detection," *Journal on Educational Resources in Computing* , vol. 4, no. 4, pp. 1–20, 2004.

[33] M. Freire, "Visualizing program similarity in the ac plagiarism detection system," in *The Working Conference on Advanced Visual Interfaces(AVI)*, 2008, pp. 404–407.

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

[35] M. Inc, "Microsoft support: How to use the windiff.exe utility," http://support.microsoft.com/KB/159214,2014/, accessed on 15.06.2019.

[36] V. Frick, C. Wedenig, and M. Pinzger, "Diffviz: A diff algorithm independent visualization tool for edit scripts," in *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2018, pp. 705–709.

[37] L. Voinea, A. Telea, and J. J. van Wijk, "Cvsscan: Visualization of code evolution," in *Proceedings of the 2005 ACM Symposium on Software Visualization*, 2005, pp. 47–56.

[38] A. Telea and D. Auber, "Code flows: Visualizing structural evolution of source code," *Computer Graphics Forum*, vol. 27, no. 3, pp. 831–838, 2008.

[39] P. Caserta and O. Zendra, "Visualization of the static aspects of software: A survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 7, pp. 913–933, 2010.

[40] A.-L. Mattila, P. Ihantola, T. Kilamo, A. Luoto, M. Nurminen, and H. Väätäjä, "Software visualization today: Systematic literature review," in *Proceedings of the 20th International Academic Mindtrek Conference*, 2016, pp. 262–271.

[41] L. Merino, M. Ghafari, C. Anslow, and O. Nierstrasz, "A systematic literature review of software visualization evaluation," *Journal of Systems and Software*, vol. 144, pp. 165–180, 2018.

[42] R. L. Novais, A. Torres, T. S. Mendes, M. Mendonça, and N. Zazworka, "Software evolution visualization: A systematic mapping study," *Information and Software Technology*, vol. 55, no. 11, pp. 1860–1883, 2013.

[43] T. Cheesman, K. Flanagan, and S. Thiel, "'translation array prototype 1: Project overview'," http://delightedbeauty.org/, accessed on 07.06.2019.

[44] T. Cheesman, "delightedbeauty.org," http://www.delightedbeauty.org/, 2011, accessed: 2017-02-16.

[45] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *Journal of the Association for Information Science and Technology*, vol. 27, no. 3, pp. 129–146, 1976.

[46] G. Salton and C. Buckley, "Term-weighting pproaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[47] 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.

[48] D. Sarkar, *Text Analytics with python*. Springer, 2016.

[49] 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.

[50] A. Huang, "Similarity measures for text document clustering," in *The Sixth New Zealand Computer Science Research Student Conference*, 2008, pp. 49–56.

[51] S. E. Robertson, "Overview of the Okapi projects," *Journal of Documentation*, vol. 53, no. 1, pp. 3–7, 1997.
[52] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, 2004.
[53] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Information Visualization*, vol. 10, no. 4, pp. 289–309, 2011.
[54] R. C. Roberts, L. McNabb, N. AlHarbi, and R. S. Laramee, "Spectrum: A C++ Header Library for Colour Map Management," in *Computer Graphics and Visual Computing (CGVC)*, G. K. L. Tam and F. Vidal, Eds. The Eurographics Association, 2018.
[55] T. Hogan, U. Hinrichs, and E. Hornecker, "The elicitation interview technique: Capturing people's experiences of data representations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2579–2593, 2016.
[56] A. Roos and T. Cheesman, "Version variation visualization (vvv): Case studies on the hebrew haggadah in english," *Journal of Data Mining & Digital Humanities*, 2017.
[57] N. Boyles, "Closing in on close reading," *Educational Leadership*, vol. 70, no. 4, pp. 36–41, 2012.
[58] C. han Jong, P. Rajkumar, B. Siddiquie, T. Clement, C. Plaisant, and B. Shneiderman, "Interactive exploration of versions across multiple documents," 2008.
[59] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann, "On Close and Distant Reading in Digital Humanities: A Survey and Future Challenges," in *Eurographics Conference on Visualization (EuroVis) - STARs*, 2015.