Automated Alignment of Medieval Text Versions based on Word Embeddings

Christofer Meinecke*, David Joseph Wrisley‡, Stefan Jänicke†

*Image and Signal Processing Group, Institute for Computer Science, Leipzig University, Leipzig, Germany
E-mail: cmeinecke@informatik.uni-leipzig.de

‡Digital Humanities, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates
E-mail: djw12@nyu.edu

†IMADA, University of Southern Denmark, Odense, Denmark
E-mail: stjaenicke@imada.sdu.dk

***Abstract*—Medieval textuality is characterized by instability in text structure and length that varies according to the text tradition. This instability in the versions, otherwise known as "mouvance", is characterized by dialectal difference, traces of orality, the modification of wording and even the rewriting and rearrangement of large parts of the text. To help humanities scholars in the exploratory analysis of such complex text collections, the visual analytic system iteal was initially proposed. The system aligns similar phrases on a line-level on the basis of string similarity and word n-grams. We propose an extension of this system that replaces the parameter-based approach with an automatic one using word embeddings thereby adding a semantic component. The benefit of the new visualization system is shown through a comparison of different versions of medieval French texts. Additionally, a domain-expert compared the parameter-based approach with the approach based on word embeddings to outline the similarities and differences in the alignments.**

***Index Terms*—Sentence Alignment, Word Embedding, Visualization, Digital Humanities**

I. Introduction

Textual scholarship focuses on the repeated reading and analysis of different versions of a text in the interest of understanding its genesis and evolution. This process can be supported by tools like TRACER [1], Versioning Machine [2] and Juxta Commons [3]. These tools are generally applicable for the comparison of versions in modern languages that exhibit minor variation, but for ancient and medieval versions, this is not always feasible. The reasons for these difficulties are the absence of sufficiently large corpora to train models for vernacular medieval languages and the instability in structure and length of the text versions. Additional reasons include the relative absence of cross-dialect lemmatization lists for these languages and their variant forms over the centuries that render methods and tools like the above-mentioned TRACER not feasible. With iteal [4] a language-independent visual analytics system exists that computes and visualizes an alignment of two versions of a text using word n-grams and string similarity, without supporting semantic similarity. We propose an automatic approach to tackle this limitation. For this, a pipeline based on word embeddings is developed to align medieval text versions on a line-level to help humanities scholars in the textual criticism process. In this context, an alignment includes a semantic or morphological match between two sentences in different text versions, visualized as a line connecting the sentences. A problem for the automatic approach is the rather small corpus for variant medieval text versions and the absence of a general language model for these language variants. To overcome these problems the proposed architecture preprocesses a "monolingual" domain-specific corpus to train a model, representing the words and sentences in a high dimensional vector space. Then a user can compare two text versions with each other with a dual view: a Distant Reading view showing the overall structure and the aligned patterns and a Meso Reading view showing both text versions in their immediate context. For each sentence, the most similar sentences that are over a specific threshold are visualized as an alignment in both views. The sentences can be further compared using a Close Reading view using Variant Graphs and a heat map showing the similarity of the word vectors. The contributions of this paper are threefold 1) the implementation of an automatic alignment pipeline, 2) the enrichment of the Close Reading view with information about the word vectors to give a domain expert insight into the "blackbox" 3) a visualized comparison of the parameter-based and the automatic approach.

LEVIA'19: Leipzig Symposium on Visualization in Applications 2019

II. Fundamentals

A. Domain-specific Background

In the humanities text is an essential knowledge source. Before the invention of printing, texts were copied by hand in manuscripts and the language in them bears the marks of elements of an oral culture. There exist multiple problems when a scribe would copy a work, for example, common writing errors like orthographic errors or "eye skip" and the omission missing of some words. More commonly, sometimes the authors added or removed parts, exercising their poetic license, changing the meaning of passages or simplify parts of the text. Textual scholarship is a specific application area of text reuse that deals with the comparison of the wide variation of such texts. Some scholars of textual criticism attempt to find

Fig. 1. Mouvance in the Oxford and Venice 7 manuscripts of the Song of Roland.

or to construct an archetype of multiple text versions, others prefer to analyze the variance, the "mouvance" across the whole tradition as evidence of the text's reception in different contexts. The most popular example is the Bible but there are other use cases like the "chansons de geste". Regardless of the philological approach, our system is useful inasmuch as it allows a corpus of similar text versions to be explored, allowing a user to find similarities and differences between them. For medieval text versions, especially for vernacular literature, Lachmann's archetype method can prove to be quite troublesome. Vernacular medieval literary texts are often authorless and dating is uncertain. The biggest problem for textual criticism is the "mouvance" of these texts. Mouvance is a term introduced by the medievalist Zumthor [5]. It addresses the instability of medieval text variations that emerge through the above-mentioned elements of an oral vernacular culture. Two effects of "mouvance" can be seen in Figure 1. Figure 1 a) shows changes in the word order of two alignments through transpositions of words in the Song of Roland and Figure 1 b) shows the rearrangement of a part and the difference in length, which creates structural differences. The problems of "mouvance" and methods to tackle them were also outlined by Jänicke and Wrisley [6].

B. iteal

For a better comparison of two textual versions, the iteal system was proposed by Jänicke and Wrisley [4], which can be seen in Figure 2. The basic idea of iteal is to compare the versions next to each other with the help of Variant Graphs [7] and a Stream Graph [8]. Visualization is used as a speculative process to highlight possible alignment candidates. The Distant Reading view is a minimap, which gives an overview of the textual versions and the connections between the phrases i.e the single line alignments. The Close Reading view shows the plain text of a phrase and its variations together with a word-level alignment, which is achieved with TraViz [9]. To combine the best of both worlds, an interactive Meso Reading view (zoomed out version of the distant reading view) was designed as an Alignment Graph. The user can read the textual versions and can identify the aligned phrases of a phrase of interest. Through a heat map array, a distant reading of the word-level alignment is shown next to each sentence. The different parameters of the system are minimum String similarity, minimum coverage, the number of matching word n-grams and an option for counting broken n-grams. The String similarity is based on the Levenshtein distance and is a threshold for how similar two words need to be in order to align. The minimum coverage is a threshold for the minimum number of words that align. The word n-grams parameter defines the minimum number of shared n-grams i.e. the largest sequence of word matches between two sentences. The broken n-gram parameter is optional and allows words from a broken n-gram sequence to count in an alignment. All these parameters account for orthographic variance but neglect the semantic component, which can be introduced through word embeddings.

C. Word Vectors

Mikolov et al. [10], [11] proposed the word2vec Skip-Gram model with Negative Sampling, which can be applied to generate a vector space representation of a given corpus without supervision. Such log-bilinear models require less time and resources to train than complex "Deep Learning" architectures like LSTM networks. The latent space, the words are projected on, has the property that similar words (words appearing in similar contexts) tend to be "near" to each other in this space. This approach can be further expanded through a method that uses subword information like fastText [12]. For a given target word and its context, the character n-grams of the target word are included. The final vector representation of a word is the average of the word vector and the n-gram vectors. The advantage of this approach is the consideration of morphology, standing in for a lack of lemmatization. Morphology studies the smallest grammatical unit of words – the morphemes. Words that share the same morphemes tend to have similar meanings. Because of this, word forms and rare words are better represented if some of their morphemes are in the training corpus. The combination of Morphology (internal information) and Distribution Theory (external information) balances out the disadvantages of either approach. A n-gram-based morphological approach can only connect words that share one or many morphemes and a distributional model cannot create a good representation for rare words or word forms. Another benefit of internal information can be seen when working with text sources that were created with Optical Character Recognition (OCR). Because of the error susceptibility of OCR, a method that can deal with orthographic variances can result in a better corpus representation. We applied the fastText architecture to

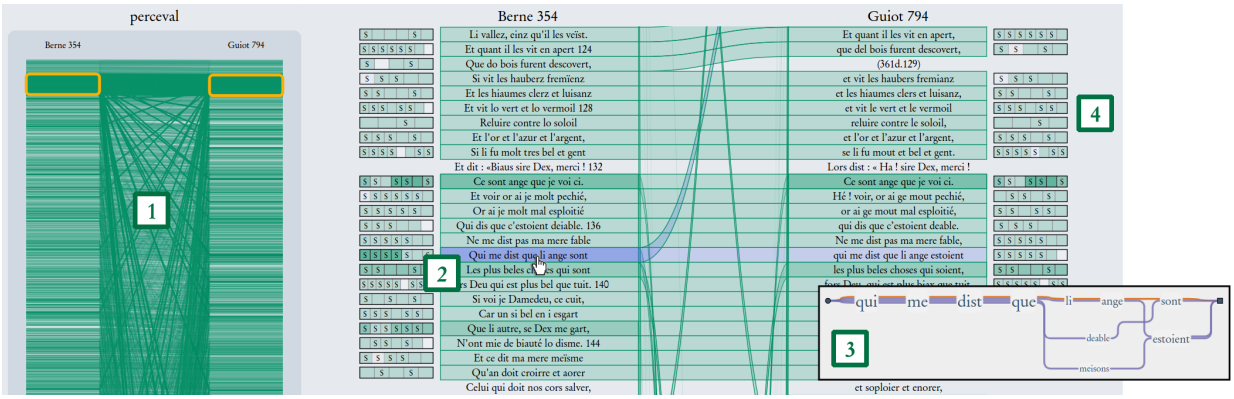

Fig. 2. An excerpt of the iteal system (1) the Distant Reading view, (2) the Meso Reading view (3) and the Close Reading view [4].

introduce semantic information into the alignments while not neglecting the orthographic variance.

D. Sentence Vectors

A fast and simple solution to create a sentence vector out of word vectors is the computation of a mean vector. A drawback to this method is neglecting word frequency, word order and the distribution of the word vectors. To further enhance this concept, weighting methods like Unsupervised Smooth Inverse Frequency (uSIF) [13] can be applied. In this weighting scheme, the assumption is made that at least every n steps a word is produced randomly, which is also the average sentence length. This assumption is used to compute the number of words always produced by chance i.e. all words with a higher frequency than the probability that a word is produced by chance for a discourse ("what is being talked about" [14]). When the average sentence length increases, fewer words are produced by chance and the consequence is that the preference of low frequent words decreases. To further improve the sentence vectors, they are combined to a matrix with each vector as the columns to apply a common component removal on all vectors. For this, the first m components will be removed as a denoising objective. Removing more components increases performance as more variance is included but the default value is chosen as $m = 5$, which is a trade-off between computational cost and performance. This method outperformed, with a much smaller dimension and without supervision, multiple supervised approaches based on recurrent neural networks for sentence similarity and sentiment analysis. Following Wieting et al. [15], who showed that complex methods to generate sentence vectors or vectors of larger parts of texts can be easily outperformed by word vectors retrained on domain data. An approach that can be combined with the previously mentioned methods is the use of Power-Mean embeddings [16], which are a concatenation of different mean values of the vectors. The idea behind this concept is that each mean value encodes different information and by using multiple means a better representation is reached. We applied a concatenation of the arithmetical mean, the uSIF weighting scheme, and the minimum and maximum.

III. Related Works

Our work includes semantic information into a visual analytic system, which should provide a more nuanced, and more accurate process for exploring alignments of textual traditions. For other application areas of text reuse, different semantic methods were proposed. For the detection of short text passages, Kusner et al. [17] used word embeddings together with different features to train a supervised classifier. For the same usage, Zhang et al. [18] applied the Fisher kernel to create text vectors with a fixed size out of a bag of word vectors. Methods based on Shingling instead of word embeddings are proposed by Smith et al. [19] and Seo et al. [20]. All these approaches did not include character information, which poses a problem for vernacular literature on account of the orthographic variance. Hazem et al. [21] combined parameter-based methods like String similarity and the Jaccard coefficient with word embeddings to detect text reuse in devotional texts of the middle ages. They noticed that different methods are good for different kinds of alignments like permutations, inflections or lexical substitutions. But they only applied pretrained embeddings and it is unclear if they included character information. For plagiarism detection Gharavi et al. [22] combined similarity measures based on word embeddings together with the Jaccard coefficient, while Zubarev and Sochenkov [23] applied contextual models like BERT to detect translated plagiarism cases. While the former method needs tuning of multiple parameters the latter approach depends on a supervised classifier. Multiple works designed different visualizations for text reuse. Similar graph-based visualizations were applied to visualize plagiarised text passages [24], and to visualize similarities and dissimilarities in 24 versions of the Bible [25]. In addition to graphs, pixelmaps [26] and heat maps [27] can be used to visualize text reuse patterns.

IV. Data

In order to compare the parameter-based approach and the automatic approach, the same medieval French texts were used as usage scenarios for both systems. The largest orthographic variance and "mouvance" of our vernacular literature corpus

is found in the Song of Roland. The length of an edition of the Song of Roland can vary from 2000 lines up to 8000 lines. The alignment process gets even more difficult through the transpositions of whole paragraphs, creating large structural differences. Because of this variance, the classification of an alignment sometimes comes down to a matter of interpretation. All these problems underline why a manual collation of these text versions is a near difficult task and an automatic approach encounters numerous problems. For the Song of Roland single-manuscript editions were employed: the Oxford manuscript (4002 lines), the Venice 7 manuscript (8002 lines) and the Lyon manuscript (2392 lines). On account of the large differences across the versions of the Song of Roland, for the sake of comparing two different versions of Chrétien de Troyes' romance "Perceval: le conte du Graal" were also used – the Berne (9494 lines) and the Guiot (9167 lines) manuscripts. The interesting feature of the romance versions is the high similarity between them and the small difference in the structure and length, which gives an easy overview of the quality of the alignments. The reason for this is Chrétien's romances are in general more stable than works of epic poetry. As an additional use case five different versions of "La vie de saint Marie l'Egyptienne" and "Vie de saint Alexis" were used. Both works are lives of saints (hagiographies) and were retold in very different styles. These versions differ in their meter and length. For both hagiographic texts, there are versions with multiple hundred lines and some versions with over 1000 lines. As additional training data, a collection of epic, historic, romance and hagiography text versions were added together with a corpus of "chansons de geste" [28]. The full medieval French corpus has a total of 1.723.922 tokens and 82.800 types.

V. Methodology

In the following, the needed preprocessing steps and the algorithmic pipeline in Figure 3 are explained. Before a model can be trained, the data need to be cleaned. The preprocessor removes all diacritics (not present in medieval language anyway, but added by editors), unnecessary white spaces, and OCR artifacts before the text is lowercased and tokenized. Part of Speech Tagging and lemmatization were excluded because there is no adequate French lemmatizer or part of speech tagger that is robust for all periods and dialects, which is a problem when dealing with some medieval vernacular literature. Another reason why no lemmatizer was used, is due to the fastText architecture. Through the character vectors, different inflections of a word are close in the latent space. After the preprocessing a monolingual corpus is fed into the gensim fastText Skip-Gram implementation [29] to train a neural network. The resulting vectors have a dimension of 100. Although there exist pretrained models for over 157 languages [30], there are two main reasons why embeddings were computed from the domain-specific corpus. One reason is the absence of models for medieval dialects, the other reason is that pretrained embeddings are trained for a wide range of tasks on a large corpus to be as universal as possible. This is not necessary for domain-specific tasks. Embeddings trained on a domain-specific corpus are known to be able to outperform the generalized embeddings for such tasks [15]. When the training of the network has finished, the word vectors can be extracted and normalized. For the use of uSIF, the frequency of every word in the text corpus is computed. Then two versions of a text of interest are compared. First, for every line or sentence, a sentence vector is computed out of the word vectors (Figure 3 A). In this case, a sentence vector is the aggregation of the arithmetical mean, the minimum, the maximum and the uSIF weighted vector, which results in sentence vectors with a dimension of 400.

A. Nearest Neighbor Search

The sentence vectors are added to an index structure based on Voronoi cells using faiss [31] (Figure 3 B). faiss is a python library, which allows searching on a large vector set in a fast way. For each centroid of a cell, a hash value is computed, which is used to access the cell in $O(1)$. The neighborhood of the centroid is defined through a distance measure. As a measure, the cosine similarity can be used to query the k-nearest neighbors. For each text version, an index is constructed and the other version is queried. Through this preprocessing step a list of potential candidates for each sentence in both versions can be generated for which the Word Movers Distance (WMD) [17] is computed (Figure 3 C). This preprocessing step is used as a speedup because the WMD is computationally expensive with a complexity of $O(p^3\ log\ p)$ with p being the number of unique words, and therefore the computation for both text versions would consume a lot of time. In contrast to the cosine similarity, the WMD is using the word vectors to compute the similarity of two sentences, which is the cumulated Euclidean distance of the words in both sentences. For a sentence d the minimum cost is computed to move it to the exact position in the d-dimensional space another sentence d' is located. Through this, the relation between semantic similar words like synonyms should be better included in the overall distance between the sentences. An example is seen in Figure 4.

$$sim(s,q) = 1 - \frac{WMD(s,q)}{maxD} \tag{1}$$

The computed WMD distance between a sentence s and a sentence q is converted into a similarity measure as seen in Equation 1. $maxD$ is the highest WMD value observed, which is used to convert the distance to a similarity measure.

B. Alignment Decision Process

To define a threshold, the average similarity between each sentence and its nearest neighbor is computed according to Equation 2 (Figure 3 D). The $kn(i,j)$ function returns the j-th nearest neighbor of sentence i and D is the set of all sentences in both versions. The assumption in using the average as the base of the threshold is that both versions of interest are connected i.e. they share the same rough archetype or they are modified versions of each other that share parts of their

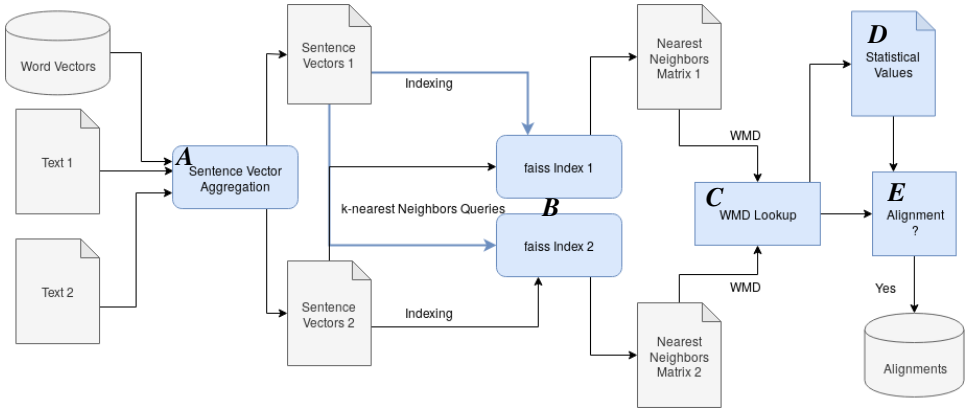

Fig. 3. An overview of the algorithmic pipeline after preprocessing and training.

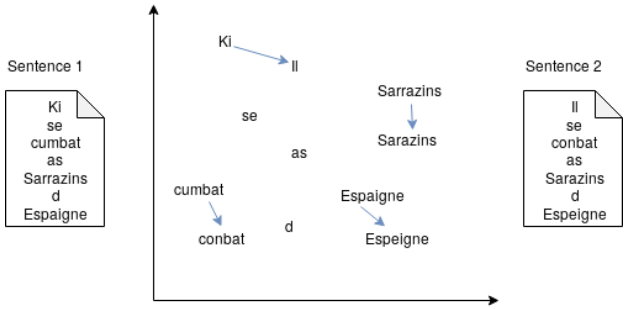

Fig. 4. The Word Movers Distance for two sentences in the Oxford and Venice 7 manuscript of the Song of Roland.

structure. The average of two very dissimilar versions would be very low and therefore not a good base for an alignment decision process. The average difference in the similarity between the first nearest neighbor and the second nearest neighbor is computed similarly according to Equation 3. Both values are used together with the difference in similarity between the current sentence q in the nearest neighbor list and his successor in the list $succ(q)$ (Equation 4) to define a threshold for an alignment. The full alignment decision threshold t is seen in Equation 5. If the similarity of the two sentences is greater than the threshold it is classified as an alignment (Figure 3 E). The parameters α and β can be added to control the influence of Equation 4 and the final penalty term. The main idea behind the terms is to allow the classifier to detect alignments with a larger variance when the difference to the other neighbors is high and to ignore edge cases where the neighbors are very similar. A reason for this can be a bad positioning of rare or unrelated words in the latent space. If $\beta\ curD$ is smaller than the $avgD$ t increases and in the other case t decreases. The α value controls the extend of the increase or decrease, while β controls the deviation from the $avgD$ term.

$$avgNN = \frac{\sum_{s \in D} sim(s, kn(s,1))}{\mid D \mid} \tag{2}$$

$$avgD = \frac{\sum_{s \in D} sim(s, kn(s,1)) - sim(s, kn(s,2))}{\mid D \mid} \tag{3}$$

$$curD = \ sim(s,q) - sim(s, succ(q)) \tag{4}$$

$$sim(s,q) \geq t$$
$$t = max(avgNN - \alpha\ (\beta\ curD - avgD), 1) \tag{5}$$

C. Visualization of the Word Movers Distance

After the computation of the alignments, the text versions can be compared within the iteal system. The different user-defined parameters were replaced by a single parameter for the WMD. The similarity slider starts at the minimum similarity value of a detected alignment. Additionally, the $avgNN$ value is displayed next to the slider, which can give a good starting point for the exploration. To see high similar patterns a good tactic is to reduce the similarity in small steps from the highest value. The Variant Graph in the Close Reading view is enhanced through arrows to indicate the word transportation of the WMD between two lines, additionally a heat map that shows the similarity of the words that are appearing in both lines was added. Both can be seen in Figure 5. The saturation encodes the similarity using a linear color scale. Total dissimilarity (a value of 0) is encoded as white and total similarity (a value of 1) is encoded as green. All values in the range $(0, 1)$ are mapped to the corresponding green-tone. Because the arrows can increase the visual clutter of the Variant Graph, they can be disabled. The heat map places the words of the first sentence on the x-axis and the words of the second sentence on the y-axis. The word transportation of the WMD is communicated through a solid border while the nearest neighbors of each word are displayed as striped squares. Through the heat map, it is easier to detect the words that are similar in each phrase and so to understand why the lines are similar in the vector space.

VI. Usage Scenarios

A. Perceval

An interesting feature of Perceval is the high similarity of the Berne and Guiot manuscripts. In Figure 6 a) an excerpt of the alignment of both versions with the parameter-based

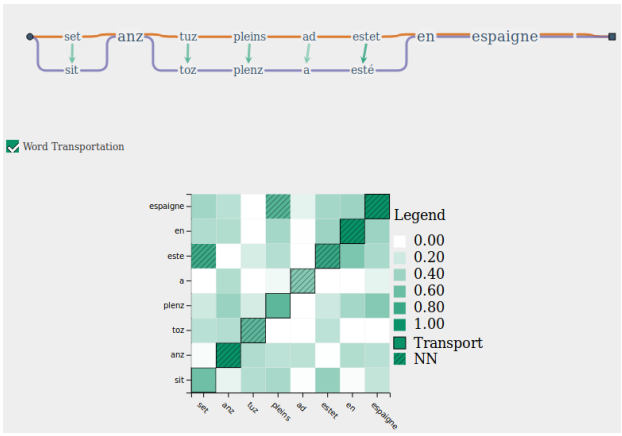

Fig. 5. The new Close Reading view, which communicates the word vectors relation and the WMD of two sentences.

approach can be seen. Figure 6 b) shows the same excerpt using the automatic approach. The whole Distant Reading view of the automatic approach can be seen in c). Except for a few insertions in the Berne and the Guiot manuscripts, which can be seen as white spaces in the Distant Reading view, both versions look perfectly aligned. When comparing to the Distant Reading view of the parameter-based approach in Figure a) fewer cross-connections are observed. A direct comparison of the Meso Reading View in Figure 6 a) and b) shows that the automatic approach cannot find all alignments and therefore a smaller recall but fewer errors are reported.

B. Song of Roland

Analog to the previous paragraph, Figure 7 a) shows an excerpt of the alignment of the Oxford and the Venice 7 manuscript with the parameter-based approach. Figure 7 b) shows the same excerpt using the automatic approach. The automatic alignment process for epic poetry like the Song of Roland is more challenging than the alignment of a stable romance like Perceval. The reasons for this are the structural difference and the “mouvance”. When comparing both Distant Reading views it can be observed that the automatic approach directly shows reused patterns, which would not be possible out-of-the-box for the parameter-based approach.

C. La vie de saint Marie l’Egyptienne

In Figure 8 a) an excerpt of the alignment of the anonymous Renart le Contrefait and the Rutebeuf manuscript with the parameter-based approach can be seen. Figure 8 b) shows the same excerpt using the automatic approach. When comparing both approaches it can be seen that the results are similar and although the string similarity approach finds some alignments that the vector space approach cannot find and the other way around, the vector space approach can find all the alignments that were added by the user (yellow lines). For the different versions of the Vie de saint Marie l’Egyptienne some parts of the story were reused, which can be detected through the visualization.

D. Domain Expert Feedback

The collaborating humanities scholar tested the automatic approach. Before testing the system he thought that since the training was done with the Camps corpus of “Chanson de geste” that there will be a bias toward alignments in the Song of Roland, because of the similarity in the structure of the poetry. This underlines the need for separate training corpora and models for each genre. When the medievalist looked at the alignments that were generated with the word embedding approach he noticed that they were often more ”synonymic” than they are ”orthographic” meaning that the alignment can be sometimes more sense driven, as if a poet were remembering not just the words memorized in order, but the general sense and was redacting them differently. When imagining that someone says “And here approached the king over the hill” and the line “The ruler then sallied forth over the ridge” is detected as an alignment, then this would not be considered an alignment in old fashioned synoptic editions, but instead would be variants. Whereas king and ruler are synonymous and hill and ridge are too, the verbs are actually opposites. The word embedding approach is perhaps in this respect better for such oral poetry that is less bookish (and copied as such), in other words, better for epic and saints’ lives than for romance. This is perhaps not a problem when the alignments are created by verbs or nouns that correspond to a general lexical field of the text, but when they are created by function word proximity in vector space, the resultant alignment is odd. One of the important things to note about the Saint Mary the Egyptian texts is that there are two very different intertextual scenarios at play. First, there is the Rutebeuf version and the Renart le Contrefait, which is a direct borrowing of the former carried out in a very ”written” way, meaning that there was a respect of the original text as if there was copying. In such situations, the top line similarity includes usually the closest lines in terms of literal meaning. These alignments would have been picked up as very strong ones also in the parameter-driven approach. As one descends the list of similarity it can be difficult to understand on what basis the alignment is made. With the other versions, we encounter more of an oral (and chanson de geste-like) situation: many broken n-grams, some similar sentence structure, some vague echoes between versions. This is particularly acute when comparing the versions AlexisOctP and AlexisP11 of “Vie de saint Alexis” where there is a different number of syllables per line as can be seen in Figure 9. One of the ways the medievalist thought the automatic approach might work better than the user parameter-driven one is in the case of hemistiche (half-line) alignments, where one has strong alignment across half the line and, that would be non-aligned across the other half with the parameter-based approach. He was also quite fascinated by the high saturation lines of poetry (those that are very connected to others) in the word embedding approach. These seem to be the lines that have high frequencies of formulae or function words. Another explanation for this could be that the word vectors for some words are not well separated

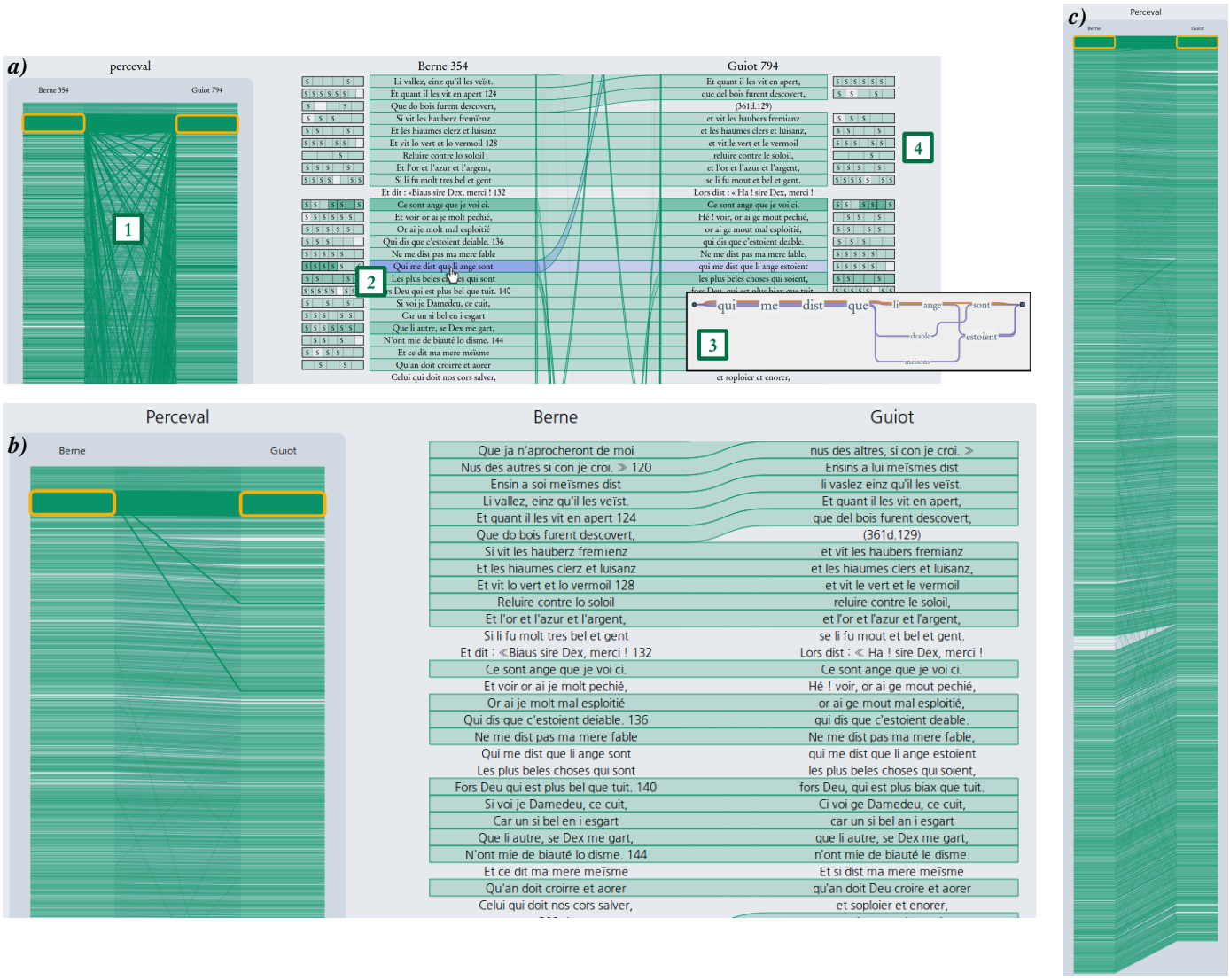

Fig. 6. Comparison of two versions of Perceval with the parameter-based approach a) [4] and the automatic approach b). Figure c) shows the resulting Distant Reading View of the automatic approach.

in the latent space, which could be improved through annotated training data or user feedback. He also mentioned, that as in the previous model, Perceval is not the best case study since it is a case of large similarity and small amount of difference.

VII. Limitations and Future Challenges

The extension from a simple string similarity model to a vector space model has multiple benefits that were highlighted in the previous sections. But still, there exist new problems and possibilities to further improve the model and the whole process. When comparing the parameter-based approach and the automatic approach it is noticeable that both have different advantages and drawbacks. While the parameter-based approach neglects the semantic component the word embedding approach focuses less on the orthographic variances. Although morphology is included in the fastText architecture through vectors of character n-grams, there could be limits. For example, two words with an edit-distance of 1 e.g. “set” and “sit” could be farther away in the vector space than the edit distance would suggest because they do not share some character n-grams. To tackle this problem, one can investigate if including of different features of the parameter-based approach like string similarity or word n-grams can improve the results. This was also supported by the user feedback, which differentiates between synonymic alignments and orthographic alignments. The automatic process also has the shortcoming that the feedback of a domain expert is not included. A user should be able to explore the text editions, scoring the results to add new alignments and remove false positives. Through this feedback, the word vectors could be updated, so that further potential alignments and false positives can be detected. As a result of this humanities scholar in the loop process, a better vector model might be generated. Similar to reinforcement learning, which uses feedback or a

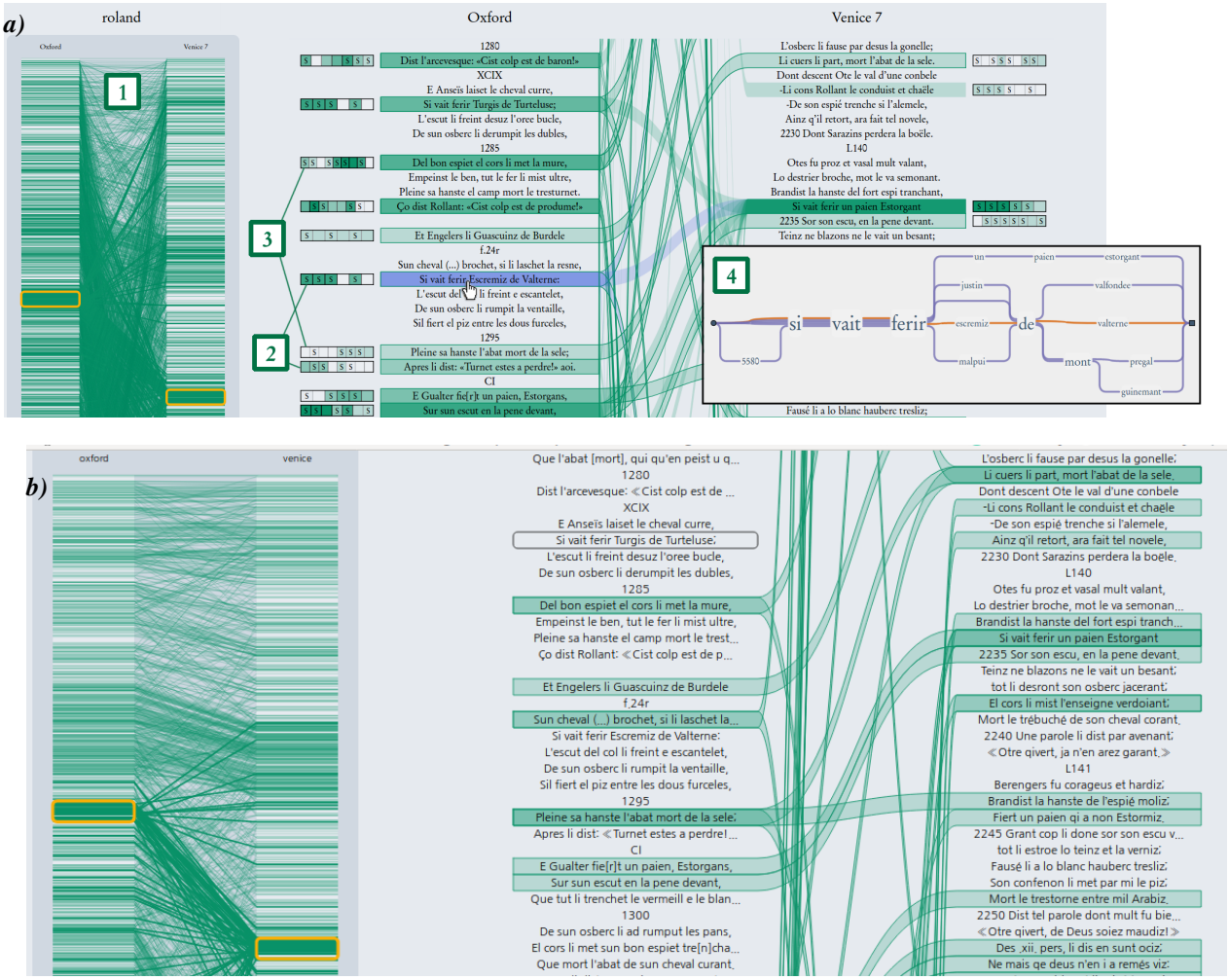

Fig. 7. Comparison of two versions of the Song of Roland with the parameter-based approach of the iteal system a) [4] and the automatic approach b).

reward from an environment and a mix of exploration and exploitation, which can lead to a reward in the long run. Another possibility to add supervision to the model is a second training step with annotated training samples. These samples could include true alignments and false positives, which can be used to train a classifier. Both types could be either constructed manually or automatically through a comparison of the parameter-based and the automatic approach. This could also help in replacing the threshold decision process, which is currently total unsupervised and based on the nearest neighbors and the distribution of the word vectors. Currently, this leads to multiple false alignments when comparing two total dissimilar text editions. Another problem the automatic approach suffers from is the small corpus size. Some words are only appearing a few times in the corpus and so their vector representation is probably not an accurate representation of their semantic. This can be amplified through potential OCR artifacts that were missed in the preprocessing step. Although these effects are reduced through the character vectors, this can still lead to a bad representation of some words or lines. The vector representation can also suffer from the polysemy of words or homonyms. A reason for this is the change of the meaning of a word over the centuries. This is a problem especially for vernacular literature and corpora, which include text from different centuries. This change can be included through a time component for the vector space model. Similar to HistWords [32], which creates different vectors for different timestamps of a word.

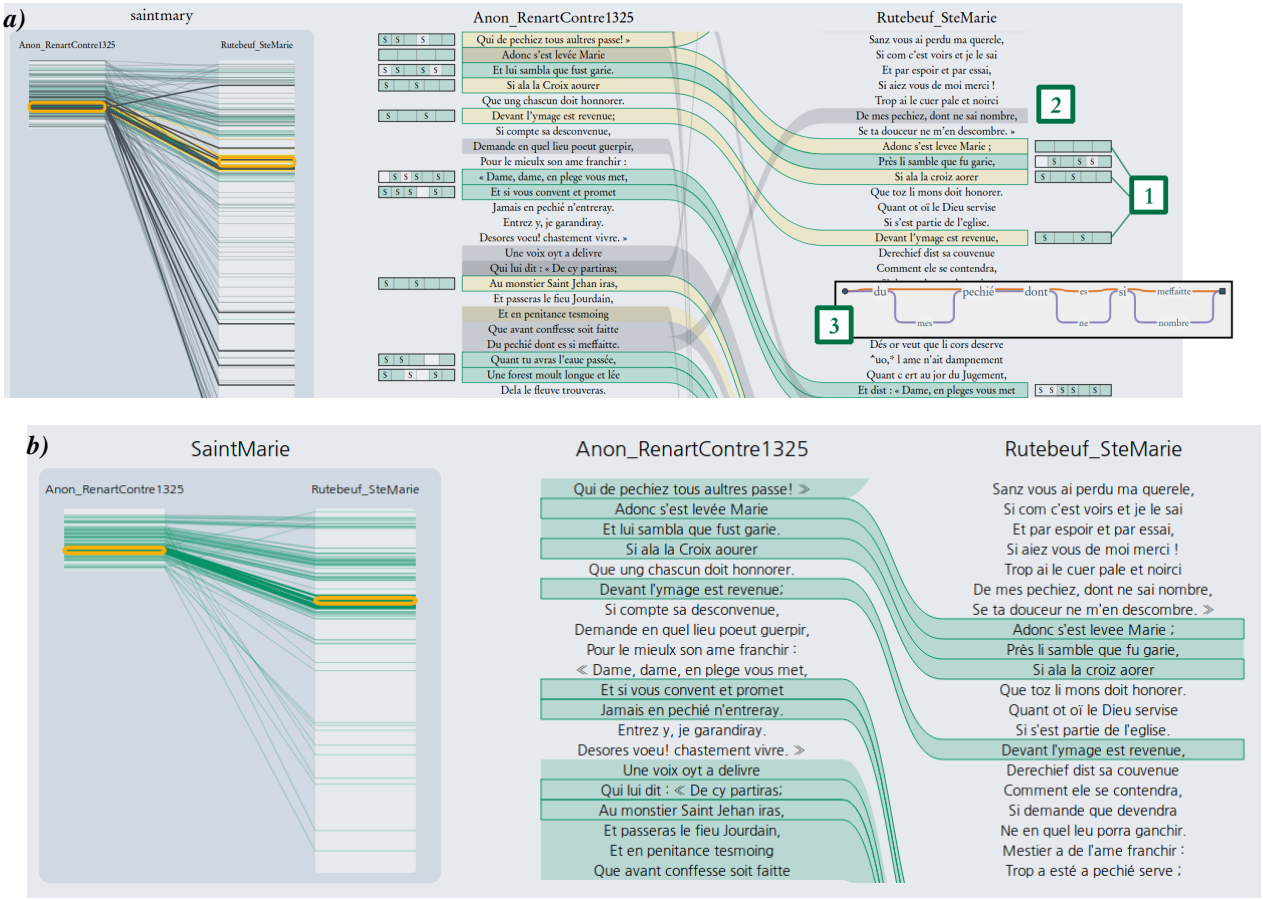

Fig. 8. Comparison of two versions of La vie de saint Marie l'Egyptienne with the parameter-based approach a) [4] and the automatic approach b). The versions are the anonymous Renart le Contrefait and the Rutebeuf manuscript.

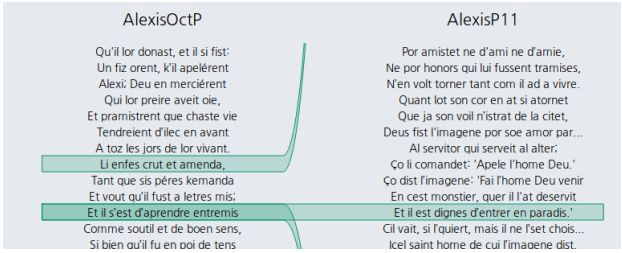

Fig. 9. An example of the alignment of the "Vie de saint Alexis" the versions are the "en vers octosyllabiques" (AlexisOctP) and the "en laisses de décasyllabes assonancés" (AlexisP11). Both versions are very dissimilar and the alignment seems odd.

VIII. Conclusion

In this work, the parameter-based approach of the iteal system was replaced with an automatic approach based on word embeddings. The system can help a humanities scholar in the exploratory workflows in textual scholarship. This was underlined with different use cases focusing on vernacular literature, especially medieval French and the feedback of a humanities scholar. Although this work focuses on vernacular literature, the approach can be easily adapted to other languages and literatures. After a preprocessing step, the fastText architecture is trained on the corpus to create a vector space representation of the corpus. Then the resulting word vectors are combined for each sentence to one sentence vector, through different methods e.g. Unsupervised Smooth Inverse Frequency and Power-Means. The normalized sentence vectors are added to a faiss index to apply a k-nearest neighbor search. The k-nearest neighbors are used as potential alignment candidates and are compared with the Word Movers Distance. For the alignment decision process different statistical features of the nearest neighbors are used e.g. the average distance to the nearest neighbors and the average difference between the distance of the following neighbors. The alignments are then visualized using three distinct views: a Distant Reading view, a Meso Reading view and a Close Reading view, the latter having been enhanced with a heat map to show the similarity of the words of the sentences in the vector space. Additionally, we presented future challenges together with the potential for improvements and further extension.

Acknowledgment

The authors thank Andreas Niekler for fruitful discussion about Natural Language Processing and Word Embeddings.

REFERENCES

[1] eTRAP (electronic Text Reuse Acquisition Project), "Tracer," "http://www.etrap.eu/research/tracer/", 2013, (Retrieved 2019-02-07).

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

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

[4] 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.

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

[6] 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.

[7] D. Schmidt and R. Colomb, "A data structure for representing multi-version texts online," *International Journal of Human-Computer Studies*, vol. 67, no. 6, pp. 497–514, 2009.

[8] L. Byron and M. Wattenberg, "Stacked graphs–geometry & aesthetics," *IEEE transactions on visualization and computer graphics*, vol. 14, no. 6, pp. 1245–1252, 2008.

[9] 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, 2015.

[10] 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.

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

[12] 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.

[13] 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.

[14] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "A latent variable model approach to pmi-based word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 385–399, 2016.

[15] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," *arXiv preprint arXiv:1511.08198*, 2015.

[16] A. Rücklé, S. Eger, M. Peyrard, and I. Gurevych, "Concatenated power mean word embeddings as universal cross-lingual sentence representations," *arXiv preprint arXiv:1803.01400*, 2018.

[17] 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.

[18] Q. Zhang, J. Kang, J. Qian, and X. Huang, "Continuous word embeddings for detecting local text reuses at the semantic level," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 797–806.

[19] D. A. Smith, R. Cordell, E. M. Dillon, N. Stramp, and J. Wilkerson, "Detecting and modeling local text reuse," in *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*. IEEE Press, 2014, pp. 183–192.

[20] J. Seo and W. B. Croft, "Local text reuse detection," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 571–578.

[21] 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.

[22] E. Gharavi, H. Veisi, and P. Rosso, "Scalable and language-independent embedding-based approach for plagiarism detection considering obfuscation type: no training phase," *Neural Computing and Applications*, 2019.

[23] D. Zubarev and I. Sochenkov, "Cross-language text alignment for plagiarism detection based on contextual and context-free models," *Computational Linguistics and Intellectual Technologies:Proceedings of the International Conference "Dialogue 2019"*, 2019.

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

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

[26] 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. Wiley Online Library, 2017, pp. 237–248.

[27] 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.

[28] J.-B. Camps, E. Albarran, A. Cochet, and L. Ing, "Geste: un corpus de chansons de geste," 2016, http://github.com/Jean-Baptiste-Camps/Geste.

[29] 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.

[30] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *arXiv preprint arXiv:1802.06893*, 2018.

[31] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *arXiv preprint arXiv:1702.08734*, 2017.

[32] W. L. Hamilton, J. Leskovec, and D. Jurafsky, "Diachronic word embeddings reveal statistical laws of semantic change," *arXiv preprint arXiv:1605.09096*, 2016.