Versus: an automatic text comparison tool for the digital humanities

Tom Wainstain
Paris Cité University
Paris, France
tom.wainstain@etu.u-paris.fr

Motasem Alrahabi
ObTIC, Sorbonne University
Paris, France
motasem.alrahabi@sorbonne-universite.fr

Abstract

Digital humanities (DH) have been exploring large-scale textual reuse for several decades: quotation, allusion, paraphrase, translation, rephrasing. Automatic comparison, made possible by the increasing digitization of corpora, opens new perspectives in philology and intertextual studies. This article presents a state of the art of existing methods (formal, vector-based, statistical, graph-based) and introduces an open-source tool, Versus, which combines multigranular vector alignment, interactive visualization, and critical traceability. This framework aims to provide a reproducible and accessible solution for DH researchers, with support for text comparison in multiple languages.

1 Introduction

Since the works of (Kristeva, 1980) and (Genette, 1982), intertextuality has referred to a variety of textual reuses: quotation, plagiarism, allusion, paratext, etc. This structural dimension of text is central to philology, genetic criticism, comparative literature, as well as to history and textual linguistics. The growing availability of digitized corpora now paves the way for large-scale automated detection (Ganascia, 2020). However, existing methods face two major limitations: a high rate of false positives in large corpora, and a weak ability to detect semantic or allusive reuse. Overcoming these challenges requires the development of tools that combine lexical alignment, semantic modeling, and critical visualization.

2 Context and objectives

This work aims to examine recent approaches and propose open-source solutions tailored to the specific needs of researchers in the humanities and social sciences. It offers a structured overview of automatic comparison tools based on the following criteria: working principle, strengths, limitations, and representative tools for each approach (formal, vector-based, statistical, graph-based). It also presents Versus, an open and reproducible tool designed to meet the specific needs of DH researchers through an interactive interface and critical traceability of results. The contribution of this work is primarily system-oriented: Versus is presented as an open-source tool that integrates and adapts existing methods for the specific needs of digital humanities, rather than as an algorithmic advance.

3 Methods and tools for text comparison

Following prior surveys of text similarity methods (Nègre, 2013); (Wang and Dong, 2020); (Prakoso et al., 2021), we adopt a four-fold categorization of approaches, which has proved useful both in DH and general NLP contexts:

3.1 Formal approaches

These approaches compare texts directly at the level of characters, words, or n-grams, without modeling meaning. They rely on measures such as edit distances (Levenshtein (Levenshtein, 1966), Hamming (Hamming, 1950)) or similarity coefficients (Jaro (Jaro, 1989), Dice (Dice, 1945), Jaccard (Jaccard, 1901)), and are effective in detecting local or superficial similarities. They are particularly suited for word-for-word comparison or the analysis of fine textual variants. Among the tools based on a formal approach, Text-Pair[1] enables the detection of similar passages—such as quotations, borrowings, or common expressions—across large text collections using sequence alignment and shingling. CollateX[2], designed for philology, automatically aligns variants within a critical editing

[1] https://artfl-project.uchicago.edu/text-pair

[2] https://collatex.net/

framework. Medite[3] uses a suffix-tree and HMM-based algorithm to align two versions of a text by detecting deletions, insertions, replacements, and displacements, supporting critical editing and textual genetics.

Passim[4], for its part, is suited to detecting textual reuse in large corpora, combining speed and robustness. Finally, Diffchecker[5] provides a simple interface for line-by-line comparison, useful for quick checks or clear visualization of local divergences.

3.2 Vector-based approaches

These transform text into vectors to measure semantic similarity (cosine, Euclidean, etc.). They are generally robust to reformulations. Two main families of vector representations coexist: lexical representations, such as BOW or TF-IDF, which count word frequencies without considering context; and distributed representations, such as Word2Vec, GloVe, or BERT, which learn vectors from usage contexts to capture semantics. Among the tools based on a vector approach, spaCy similarity[6] offers fast measurement of semantic similarity between text units, based on built-in representations. Sentence-Transformers[7] generates robust sentence embeddings, well-suited for detecting reformulations and semantic alignment. LASER[8], developed by Facebook, provides multilingual representations for comparing texts across languages. Gensim[9] offers classic models like Word2Vec and Doc2Vec, effective for capturing lexical similarities in large corpora. Finally, SimAlign[10] combines lexical alignment and contextual embeddings to identify word-level correspondences, including in multilingual settings.

3.3 Statistical approaches

These methods leverage machine learning to identify patterns of similarity across texts. Supervised models—such as DSSM, ARC-I, or MV-LSTM—are trained on annotated examples to predict textual alignment or correspondence. Unsupervised techniques—such as LSA, LDA, or clustering—reveal latent structures without prior labeling, enabling thematic grouping, topic inference, or segment classification. These approaches are particularly effective for mapping global semantic proximities and visualizing the structure of large corpora in reduced vector spaces. In digital humanities, they support the exploration of textual traditions, discursive dynamics, and stylistic variation across time or authorship. Tools like Orange Text Mining[11] offer an accessible graphical interface for clustering and topic modeling. Scikit-learn[12] provides a robust suite of unsupervised algorithms for grouping high-dimensional representations. BERTopic[13], which combines transformer-based embeddings with dimensionality reduction and density-based clustering, enables the extraction of coherent and interpretable topics from heterogeneous or multilingual corpora.

3.4 Graph-based approaches

These approaches represent texts as networks of relationships (semantic, syntactic, discursive). They model links between words, sentences, or entities using knowledge graphs or graph neural networks (GNNs). While not all graph-based tools are designed specifically for text comparison, they can contribute to similarity analysis through their capacity to model textual structures that can then be compared using graph-based metrics such as structural comparison, node centrality analysis, and subgraph matching algorithms. These methods capture the global structure of the text and are effective for analyzing complex or multi-level connections, particularly in long or structured texts.

Among graph-based tools, textnets[14] represents collections of texts as networks of documents and words, enabling comparative analysis through network visualization and structural metrics. The tm-toolkit[15] from WZB offers comprehensive text mining and topic modeling capabilities with network analysis features that can support comparative analysis of semantic structures extracted from text corpora. Gephi[16], often used in combination with

[3] https://obtic.huma-num.fr/medite/

[4] https://programminghistorian.org/en/lessons/detecting-text-reuse-with-passim

[5] https://www.diffchecker.com/text-compare/

[6] https://spacy.io/usage/linguistic-features#vectors-similarity

[7] https://www.sbert.net/

[8] https://github.com/facebookresearch/LASER

[9] https://radimrehurek.com/gensim/

[10] https://github.com/cisnlp/simalign

[11] https://orangedatamining.com/

[12] https://scikit-learn.org/stable/modules/clustering.html

[13] https://maartengr.github.io/BERTopic/

[14] https://github.com/jboynyc/textnets

[15] https://github.com/WZBSocialScienceCenter/tmtoolkit

[16] https://gephi.org/

external text analysis tools such as TXM or custom pipelines, provides a powerful solution for visually exploring complex graphs derived from textual data, particularly for exploratory analysis. TextRank[17] applies the PageRank algorithm to lexical graphs and is useful for automatic keyword extraction or summarization within individual texts. Finally, discoursegraphs[18] enables the annotation and analysis of discursive and argumentative relations in texts using enriched directed graphs for multi-level annotated corpora.

These methods are particularly effective for analyzing textual structures, though dedicated text comparison typically requires additional algorithmic layers built upon these graph representations.

4 Synthesis

These four approaches offer complementary strategies for comparing texts, ranging from fine-grained variant detection to deep semantic modeling. The choice of method depends on the type of data, the desired level of granularity, and the goals of the analysis (reuse detection, alignment, thematic clustering, etc.).

5 Presentation of the Versus tool

Versus[19] is an open-source application dedicated to automatic text comparison, designed to meet the specific needs of DH researchers. It is based on methods that combine semantic vectorization, lexical weighting, and interactive visualization. Thanks to its reliance on multilingual sentence embeddings, Versus can be applied to texts in a variety of languages without requiring language-specific adaptation. Two main modules are currently implemented: comparison of a document with a corpus, and fine-grained comparison between two texts. The software architecture relies on a clear object-oriented model: each instance of Document is linked to a Text instance, which is further divided into Sentence and Word, enabling multi-level granularity. These documents are contained within a Corpus object. Text-to-text comparisons are handled by a PairText class. Shared variables are centralized in a Global_stuff class. The user interface is built with the Streamlit library, offering a modular structure for features and ensuring both accessibility for non-technical users and reproducible deployment beyond a simple notebook setting.

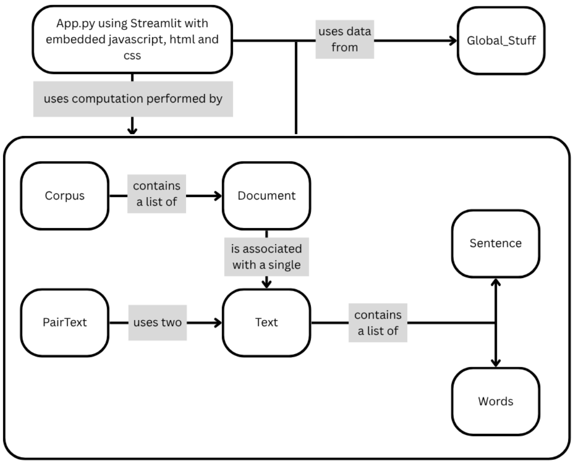

App.py using Streamlit with embedded javascript, html and css
uses data from
Global_Stuff
uses computation performed by
Corpus
contains a list of
Document
is associated with a single
Sentence
PairText
uses two
Text
contains a list of
Words

Figure 1: General Architecture of Versus

5.1 Document–Corpus Comparison

This module ranks an entire corpus based on similarity to a source document. To achieve this, Versus combines two complementary approaches: TF-IDF, used to weight the lexical importance of sentences according to their specificity within the corpus (contextual weight); Sentence Transformers (model all-MiniLM-L6-v2[20]), which generate dense 384-dimensional vector representations for each sentence. This model was selected for its balance between semantic performance, inference speed, and effectiveness on large corpora. The model operates on a general-purpose corpus (pretrained on diverse web/textual data), offering domain-agnostic performance without requiring domain-specific corpora or ontologies. Concretely, for each document, an embedding vector is assigned to each sentence using the transformer model. Each of these vectors is then weighted by the sum of the TF-IDF scores of its constituent words. This strategy gives greater value to sentences containing discriminative terms within the corpus.

5.2 Text–Text Comparison

This module is designed to identify similar passages between two texts. It uses a sliding segmentation into word n-grams, each vectorized using the same transformer model. Although the model is optimized for full sentences, it produces reliable

[17] https://github.com/summanlp/textrank

[18] https://github.com/arne-cl/discoursegraphs

[19] https://versuser-n5tyntby6aud5yryzwrdgf.streamlit.app/

[20] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Approach	Principle	Strengths	Limitations	Example Tools
Formal	Direct comparison on strings (characters, tokens, n-grams)	Simple, precise, suited to local variants	Does not capture meaning, sensitive to reformulations	Text-Pair, CollateX, Medite, Passim, Diffchecker
Vector-based	Text representation as vectors (lexical or distributed)	Robust to reformulations, partial/contextual semantics	Loss of fine-grained info, requires training corpora	spaCy, Sentence-Transformers, LASER, Gensim, SimAlign
Statistical (ML)	Learning to group or match texts	Adaptable models, useful for classification/exploration	Needs annotated data (supervised), low interpretability	Orange, Scikit-learn, BERTopic
Graph-based	Texts represented as semantic/syntactic networks	Captures complex, multi-level structures, explicit links	Complexity, high computational cost	TextNetworkX, GraphText, Gephi, TextRank, Discourse Graphs

Table 1: Comparison of text analysis approaches

vectors for word groups, including reformulated or reordered segments. The comparison is based on a cosine similarity matrix between all n-grams of both texts. To manage memory usage, the matrix is built segment by segment and converted into a sparse structure by filtering out scores below a threshold p. This ensures controlled memory usage even for large texts. The segmentation divides the second text into blocks of k columns, avoiding the creation of a full matrix, which would be memory-intensive. At each iteration, only a partial section of the matrix is computed, filtered, and converted to a sparse format, then concatenated with previous segments. This approach allows efficient processing of very large texts while limiting RAM usage. This method has proven up to 7 times faster than traditional text comparison using optimized libraries like Rapidfuzz[21], while remaining sensitive enough to detect inflected or allusive correspondences.

5.3 Interactive Visualization

The application is built on Streamlit[22] and offers an accessible interface structured into four sections: corpus management, ranking, comparison, and user guide. Users can adjust parameters (n-gram size, similarity threshold, stopword activation) and visualize detected segments in an aligned and annotated format. Results include dynamic highlighting of correspondences, difference visualization (via the Difflib[23] library), and direct interaction with the source text, ensuring readability and transparency. Correspondence with the original text is maintained through position metadata (start and end) associated with each word, allowing accurate display even when stopwords are removed. An embedded JavaScript script enables dynamic adjustment of the textual context size around matched passages. Ongoing developments include support for export in CSV and TEI formats.

5.4 Use Case Scenario

A typical use case involves comparing a source document to a collection of documents. The tool ranks the collection based on a similarity score computed from a weighted average of sentence vectors (TF-IDF + Sentence Transformers). The user selects a document, which is then aligned with the source document. A sliding n-gram segmentation detects similar passages using a cosine similarity matrix. The interactive interface displays the aligned texts side by side, highlights the detected correspondences, allows parameter adjustment (n-gram size, threshold, stopwords), and provides dynamic, annotated visualization of similarities and differences.

6 Evaluation

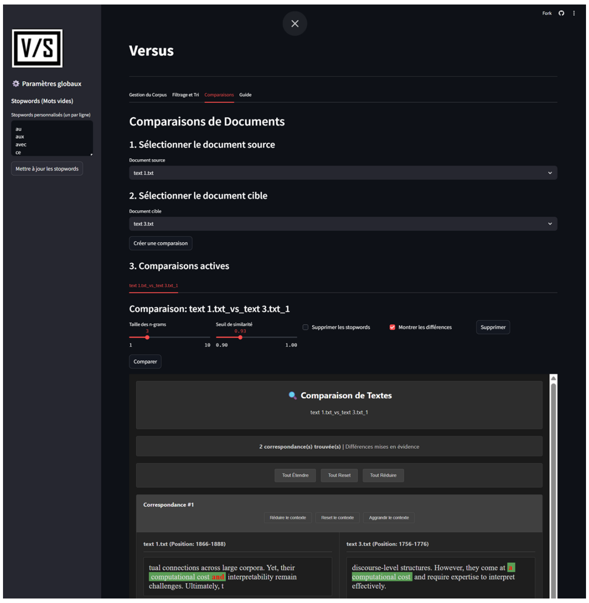
This evaluation aims to illustrate the performance and usability of the Versus tool in a realistic context of intertextual analysis.

- Quantitative: On a sample of 5 manually annotated text pairs (50 target alignments), Versus achieves an average precision of 0.86, recall of 0.79, and F1-score of 0.82. False positives mainly involve borderline reformulations or contextually ambiguous segments.
- Qualitative: The detected alignments are largely considered relevant, including in cases of reformulation or allusion. Similar segments

[21] https://rapidfuzz.github.io/RapidFuzz/

[22] https://streamlit.io/

[23] https://docs.python.org/3/library/difflib.html



Paramètres globaux

Stopwords (Mots vides)

Stopwords personnalisés (un par ligne)

au
aux
avec
ce

Comparaisons de Documents

1. Sélectionner le document source

Document source: text 1.txt

2. Sélectionner le document cible

Document cible: text 3.txt

3. Comparaisons actives

Comparaison: text 1.txt_vs_text 3.txt_1

Taille des n-grams: 3 (1–10)

Seuil de similarité: 0.93 (0.90–1.00)

☐ Supprimer les stopwords

☑ Montrer les différences

Comparaison de Textes

text 1.txt_vs_text 3.txt_1

2 correspondance(s) trouvée(s) | Différences mises en évidence

Correspondance #1

text 1.txt (Position: 1866-1888)

tual connections across large corpora. Yet, their computational cost and interpretability remain challenges. Ultimately, t

text 3.txt (Position: 1756-1776)

discourse-level structures. However, they come at a computational cost and require expertise to interpret effectively.

Figure 2: Versus User Interface

are well localized and clearly visualized, facilitating philological analysis. While tailored to DH use cases, future evaluation may consider adapting standard benchmarks such as STS-B or MSRP to literary or historical datasets.

- Ergonomic: The Streamlit interface is considered intuitive. Parameters (n-gram size, threshold, stopwords) are easily adjustable. Result export and aligned visualization provide strong support for critical analysis.

We acknowledge that this evaluation is limited in scope, relying on only five annotated pairs and lacking both baselines and error analysis, which will be addressed in future work.

7 Limitations

This work has several limitations. First, the evaluation is based on a small proof-of-concept dataset (5 annotated pairs), without systematic baselines or detailed error analysis, which restricts the strength of empirical claims. Second, Versus currently relies on transformer-based embeddings, which can be sensitive to noisy input such as OCR errors, typos, or unsupported languages. These constraints, already noted in the evaluation and conclusion, underline the need for broader benchmarking and methodological refinement, as outlined in the Perspectives section.

8 Conclusion and Perspectives

Text comparison is a central challenge in digital humanities. Versus offers a hybrid and accessible approach, combining lexical precision, semantic modeling, and critical visualization. Designed for digital humanities, it addresses key challenges in the field: processing large corpora, detecting various types of reuse (quotation, allusion, paraphrase, reformulation), ensuring result readability, and providing direct interpretive support for DH scholars without technical expertise.

By leveraging deep learning and lexical statistics techniques (transformers + TF-IDF), it enables efficient multigranular alignment, suited to the linguistic variation typical of literary and historical texts. However, this method has limitations: the transformer model relies on prior understanding of words. If the input is noisy (OCR errors, typos, unsupported languages), the resulting embeddings may be unrepresentative, reducing comparison accuracy.

As an open, reproducible, and modular tool, Versus provides a solid foundation for contemporary intertextual analysis. Unlike general NLP approaches focused primarily on model performance, our methodology emphasizes the specific requirements of digital humanities, integrating critical traceability—through alignment metadata, visualization of correspondences, and direct linkage to source texts—together with interactive visualization and accessibility for non-technical users.

Planned extensions include diachronic alignment and broader multilingual coverage, enabling cross-linguistic analysis within a unified framework. A pilot study on 18th- and 19th-century French literary texts will explore influence through paraphrase detection, forming part of a broader validation with DH scholars to assess usability, interpretive value, and alignment with philological practices. Future work will also address systematic benchmarking through larger evaluation sets and comparisons with established baselines, to better situate Versus within the state of the art.

References

Lee R. Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Jean-Gabriel Ganascia. 2020. *Les humanités numériques*. CNRS Éditions.

Gérard Genette. 1982. *Palimpsestes: La littérature au second degré*. Seuil.

Richard W. Hamming. 1950. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160.

Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579.

Matthew A. Jaro. 1989. Advances in record-linkage methodology as applied to the 1985 census of tampa florida. *Journal of the American Statistical Association*, 84(406):414–420.

Julia Kristeva. 1980. *La révolution du langage poétique*. Seuil.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Élsa Nègre. 2013. Le traitement automatique du texte pour les sciences humaines. *Traitement Automatique des Langues*, 54(1):135–152.

Dimas Wibisono Prakoso, Asad Abdi, and Chintan Amrit. 2021. Short text similarity measurement methods: a review. *Soft Computing*, 25(6):4699–4723.

Shuai Wang and Wei Dong. 2020. A survey on text similarity techniques. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(10):12–25.