Metric for Text Resources

Clemens H. Cap

January 2025

1 Background

In the collaborative development of texts (for example: Wikipedia, Open Educational Ressources, Social Coding) often the following question arises:

For developing the new version s, how much content has been used from the existing sources $s_1, s_2, \ldots, s_n$ and how much new content has been added in the process.

This question is important for numerous applications in content development:

1. For giving credit to the respective sources and authors.
2. For generating a phylogenetic tree of existing documents.
3. For tracing back the history of new ideas.
4. For identifying significant changes between documents.
5. For detecting plagiarism.
6. For tracking the influence of sources.
7. For studying knowledge diffusion in human content networks.
8. For studying the impact of a training text corpus on large language models.
9. For distinguishing editing, paraphrasing and transcribing from generating new ideas.

The application we are most interested in for the purposes of this Thesis topic is tracing new ideas via a phylogenetic tree of document versions and knowledge diffusion in human content networks.

We are not so much interested in plagiarism detection, despite the fact that this is an important recent application.

2 Initial State of the Art

There is considerable state of the art in this area. The most well known approach is, of course, LEVENSHTEIN distance $d(A, B)$ between two documents. Upon more close inspection one sees that there are numerous obstacles for a naive application of LEVENSHTEIN distance:

1. Documents may differ syntactically but could still be similar or identical in semantics. For example, one version of a document is written in HTML, the second version is written in LaTeXand the third version is written in RTF/DOC.
2. Documents may be translated versions from each other.
3. Documents may be transpositions and rearrangements of paragraphs.
4. There may be a larger set of documents, not just two documents A and B.

Additional state of the art is given by a quick-and-dirty ChatGPT search in the appendix of this document.

3 Research Questions

The task of this Thesis is to research, compare, test and develop algorithms helpful for this task.

We are not expecting a single student to solve all the issues and we look forward to defining the precise and specific question of the Thesis in every individual case. We are also open for collaborative Theses of student groups.

4 Appendix: ChatGPT Overview

A ChatGPT search turned up the following additional aspects:

4.1 Quantitative Analysis (Text Statistics)

Cosine Similarity: Measure the similarity between two text resources based on their vector representations. Useful to see how the content of one text aligns or diverges from another.

Jaccard Index: Calculate the proportion of shared words or phrases between the two texts.

TF-IDF (Term Frequency-Inverse Document Frequency): Identify the unique contributions of terms in one text and check how many of these terms are present in the other. Tools: Use libraries like scikit-learn or NLTK in Python for these computations.

4.2 Qualitative Content Analysis

Thematic Analysis: Identify key themes or ideas in both texts and evaluate how themes from one influence the themes of the other.

Sentiment Influence: If one text has a clear emotional tone, analyze how it affects the sentiment of the other (e.g., positive, neutral, negative).

Argument Analysis: If the texts are argumentative, identify whether arguments in one text support, refute, or modify the arguments in the other. Tools: Content analysis can be done manually or supported by tools like NVivo or Atlas.ti.

4.3 Intertextuality Analysis

Analyze references, quotes, or allusions in one text to the other. Use citation tracking or text search methods to identify explicit and implicit links. Example:

Direct quotes can be searched with exact-match algorithms. Paraphrases or indirect influence can be assessed using Natural Language Processing (NLP) techniques.

4.4 Dependency Analysis (Cause-Effect)

Causal Relationship: If text A causes changes in text B, track these changes (e.g., word choice, sentiment, structure).

Version Comparisons: For evolving texts, use tools like diff utilities to compare versions and isolate changes.

4.5 NLP-Based Influence Analysis

Word Embeddings: Use models like Word2Vec or BERT to represent both texts as embeddings, then compute similarity or contextual impact.

Semantic Role Labeling (SRL): Analyze how actions, subjects, or objects in one text influence those in the other.

Language Models: Fine-tune a model (e.g., GPT) on one text and see how it predicts or generates content for the other. Example: Fine-tune a language model on Text A and measure its output accuracy or creativity when applied to Text B.

4.6 Libraries

Python libraries like spaCy, gensim, and scikit-learn for text similarity and impact measurement.

Plagiarism Detection Tools: Tools like Turnitin or Copyscape to compare Wikipedia text against external content.

Diff Tools: Analyze textual changes between revisions (e.g., UNIX diff, Git tools, or Python libraries like difflib).