

Database: The Guthenberg Project (20 books from Children's Literature)

Started coding on a Jupyter Notebook to clean the dataset, split the books into paragraphs and sentences and play around with embeddings

First embedding : Sentence-BERT (cosine similarity = semantic distance)

Next step : Find the mathematical formulation for finding an ordering that minimizes the distance between neighbouring sentences (order 1) and implement it, consider higher orders ?

Implemented and tested an algorithm that finds a local minimum for the minimal distance ordering problem with circular list of sentences

To test performance, but also if we want to eventually try a full end-to-end transformer network with pages as tokens, we need a loss function for orderings  $\Rightarrow$  distance over permutations :

$$\#swaps \left( \sigma_2^{-1}(\sigma_1) \right)$$

Researched on the structure of neural networks and understood the structure of the classifier to implement : a (freeze-)BERT + concatenation to introduce an asymmetry between A and B + linear fully connected + ReLU + linear  $n$  to 1 + sigmoid

Next steps : Alternative increment algo for minimal distance ordering, find an children's books author who has some 10-20 books in The Gutenberg Project, better distance on permutations, implement the classifier