**PART: Pre-trained Authorship Representation**

1. **ARCHITECTURE**

We are trying to explain the architecture behind the PART. We propose extensive explanation from the data handling stage till the output of the comparison of two embedded documents with similarity function of the.

* 1. **Data Handling**
  2. **Standardized Gutenberg**

This dataset contains authored books, as well as analyzable meta-data such as age or book type. Books with anonymous writers or authored by multiple persons are discarded from the data set.

* + 1. **Pre-Tokenization**

The data is pre-tokenized using the tokenizer from the NLTK. This tokenizer is based on several tokenization methods such as: White Space Tokenization, Dictionary based Tokenization, Ruled-Based Tokenization, Regular Expression Tokenization, Penn Treebank Tokenization, Spacy Tokenization, Moses Tokenization and Subword Tokenization. The pre-tokenized data left some identifying information about the authors at the beginning and the end of each book.

* + 1. **Chunking**

The pre-tokenized data is being split in chunks of 512 tokens. Each chunk is then considered for training as long as the number of chunks for an individual author is 2 or more. The first and last chunk of every book is being dropped out, cause of the identify information about the author that is them.

* + 1. **Tokenization**

The tokenizer used in this model was the RoBERTa – large's tokenization algorithm – Byte Pair Encoding (BPE). The BPE is popular algorithm for subword tokenization. The main goal of this algorithm is to find a way to represent your entire text dataset with the least amount of tokens. After the tokenization the 512 tokens are transferred to the next level in the architecture.

* + 1. **Frozen Pre-trained Transformer**

RoBERTa-Large is the frozen pre-trained transformer that is being used in this model. RoBERTa has the same architecture as the BERT model and basically optimizes some hyper-parameters for BERT. This model is prepared for fine tuning for our task - Authorship Representation Embedding.

(פה נכנס הסבר על הארכיטקטורה של ברט וגם הסבר למה צריך "להקפיא" חלק מהחלקים שלו, שזה בעצם להסביר למה משתמשים במודל שכבר אומן מראש.