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 until the output of the comparison of two embedded documents with similarity function of cosine.

* 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.  
Each author document set is merged with a separator token in-between. 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, because of the identifying information about the author and the book they contain.

* + 1. **Tokenization**

The tokenizer used in this model was the RoBERTa – large's tokenization algorithm – Byte Pair Encoding (BPE). The BPE is a 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**

BERT’S architecture is based on part of the Transformer architecture – the encoder. BERT model is built by stacking up Transformer’s encoders. BERT is pretrained on 2 tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).  
In this case, we use the RoBERTa-large model that is built from 24 encoders.

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. The hyper-parameter changes made by RoBERTa are:

* Longer training: increasing the number of iterations from 100K to 300K and then further to 500K.
* Larger training data (x10, from 16G to 160GB).  
  RoBERTa is trained on 160GB of text data using 1024 32GB V100 GPUs in 1 day.
* Larger batch size (from 256 in BERT to 8k).
* The removal of the NSP task. ( pretrained with the MLM task, instead)
* Bigger byte-level BPE vocabulary size (from 30k character-level to 50k subword units).
* Longer sequences are used as input (but still keep the limitation of 512 tokens).
* Dynamic masking - Dynamically generating and changing the masking pattern applied to the training data every time a sequence is fed to the model.

These important differences, which are mostly just changing some values in the pretraining, provided a large improvement over the originally reported BERT-LARGE results.

The transformer has been frozen to preserve the transformer ability to interpret language as trained by the masked Language Modeling loss. In turn, the transformer is able to quickly produce semantic word embeddings for training from all non-padding tokens, without losing any of the original capabilities. This model is prepared for fine tuning for our task - Authorship Representation Embedding. The semantic word embeddings are a matrix with dimension (L, K) where L is the sequence length and K is the number of features. That means that the last hidden layer dimension is in size K.

* + 1. **Bidirectional LSTM (BiLSTM)**

the semantic word embeddings have to be interpreted, and for this purpose we append a bidirectional LSTM to the architecture. The BiLSTM is more efficient on lower amounts data points that a transformer layer, therefore the end representations are obtained with a recurrent network. We extract K/2 features for each LSTM pass, to form an embedding of size K.