**NATURAL LANGAUGE PROCESSING LAB**

**NAME:** Mohammed Saad Belgi

**UID:** 2021700005

**EXPT. NO.:** 9 + 10

### TITLE: Mini project

### AIM: Extracting question answering

**PROBLEM STATEMENT:**

Extractive question answering: Given a context and a question, extract answer to that question from the context

**THEORY:**

Extractive question answering is a technique used in natural language processing (NLP) and information retrieval to automatically generate answers to questions posed in natural language. In extractive question answering, the system extracts the answer directly from a given set of texts or documents without generating new text. This is typically done by identifying and selecting specific segments of text that contain the information required to answer the question.

Transformer models have proven to be highly effective for extractive question answering tasks due to their ability to capture long-range dependencies in text and understand contextual relationships effectively.

Procedure:

1. **Pretraining on large text corpora:** Transformer models, such as BERT (Bidirectional Encoder Representations from Transformers) and its variants like RoBERTa, are pretrained on large-scale text corpora using self-supervised learning objectives. This pretraining allows the model to learn rich contextual representations of words and sentences.
2. **Fine-tuning on extractive question answering data:** After pretraining, the transformer model is fine-tuned on specific extractive question answering datasets. During fine-tuning, the model learns to identify relevant segments of text within a document that contain the answer to a given question. The model is typically trained using supervised learning, where the training data consists of question-document pairs with annotated answer spans.
3. **Token-level classification:** In extractive question answering, the task is often framed as a token-level classification problem. Given a question and a document, the model predicts a probability distribution over tokens in the document, indicating the likelihood that each token is part of the answer span. The answer span is then determined based on the highest-scoring contiguous sequence of tokens.
4. **Special token representations:** Transformer models for extractive question answering often utilize special token representations to encode the question and distinguish it from the document text. For example, the [CLS] token introduced in BERT is often used to represent the question, while the [SEP] token separates the question from the document text.
5. **Post-processing:** After obtaining the probability distribution over tokens, post-processing techniques may be applied to refine the predictions and extract the final answer span. This may involve thresholding the token probabilities, considering only contiguous spans, or applying heuristics to improve answer quality.

**Dataset used:**

**S**tanford **Qu**estion **A**nswering **D**ataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable. It contains around 1,00,000 questions. All of these questions are answerable from the given context.

**Model used:**

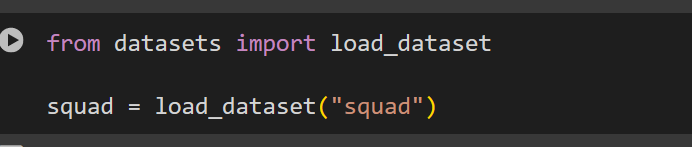
The DistilBERT model was proposed in the blog post Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT, and the paper DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than *google-bert/bert-base-uncased*, runs 60% faster while preserving over 95% of BERT’s performances as measured on the GLUE language understanding benchmark.

**NOTEBOOK LINK:**

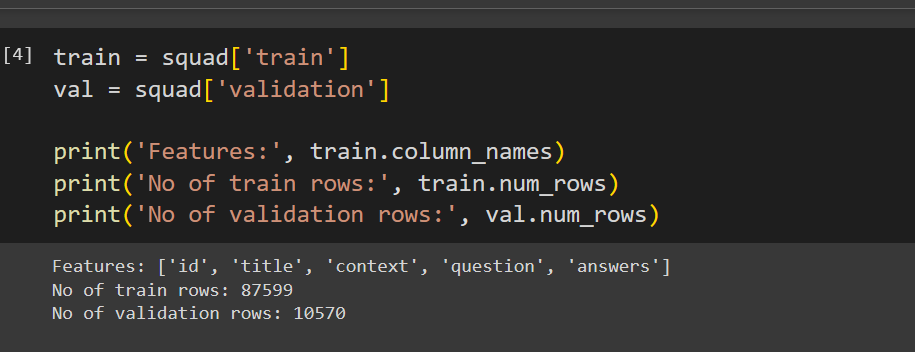
<https://colab.research.google.com/drive/1fXCv_D3_MPyS8Rd_uIplkqwza6M_5a8-?usp=sharing>

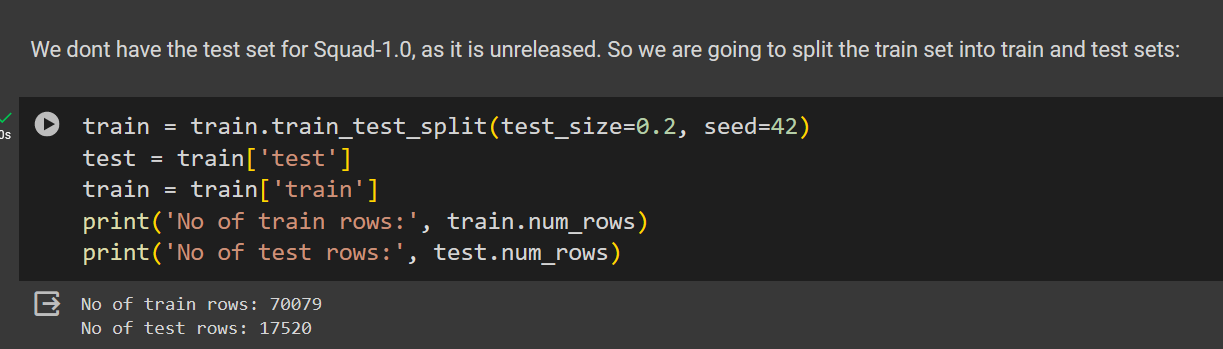
**EXPERIMENTATION AND RESULT:**

Loading the squad dataset:

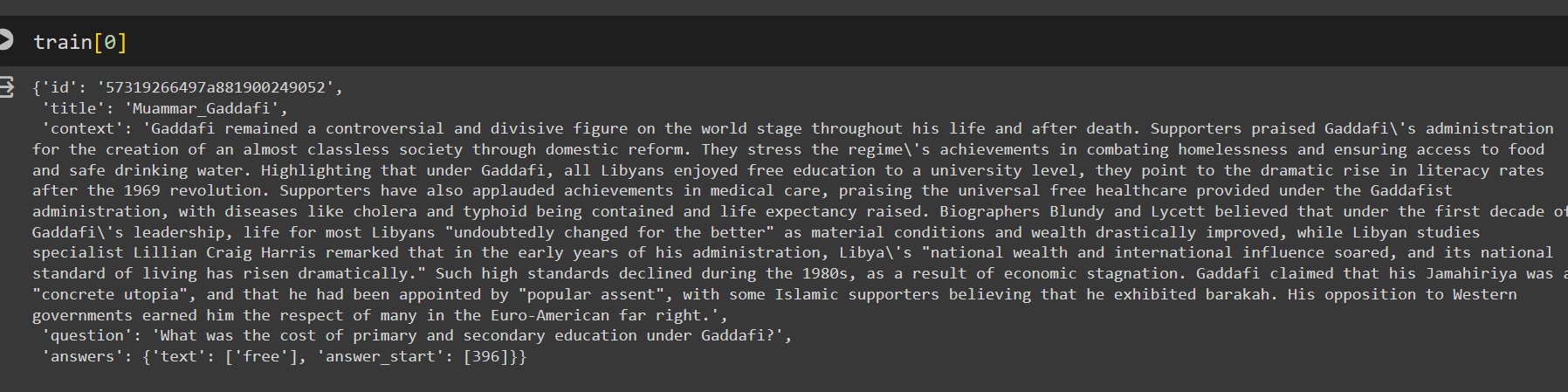


Train test and validation splits:

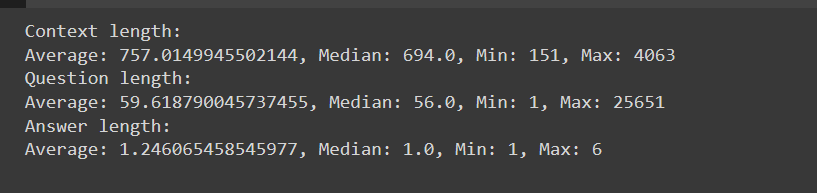




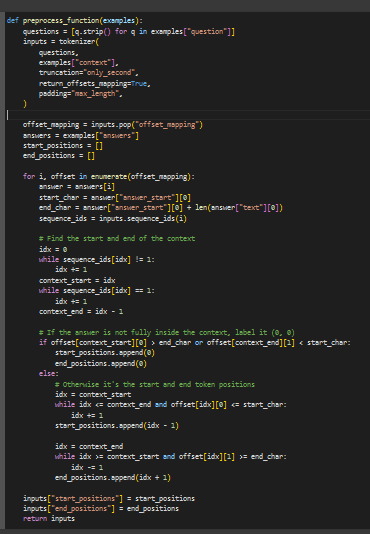
Example row:



Min, max, mean and median lengths:



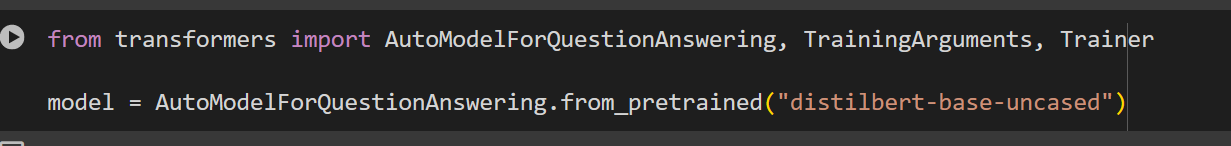
Tokenization (preprocessing):

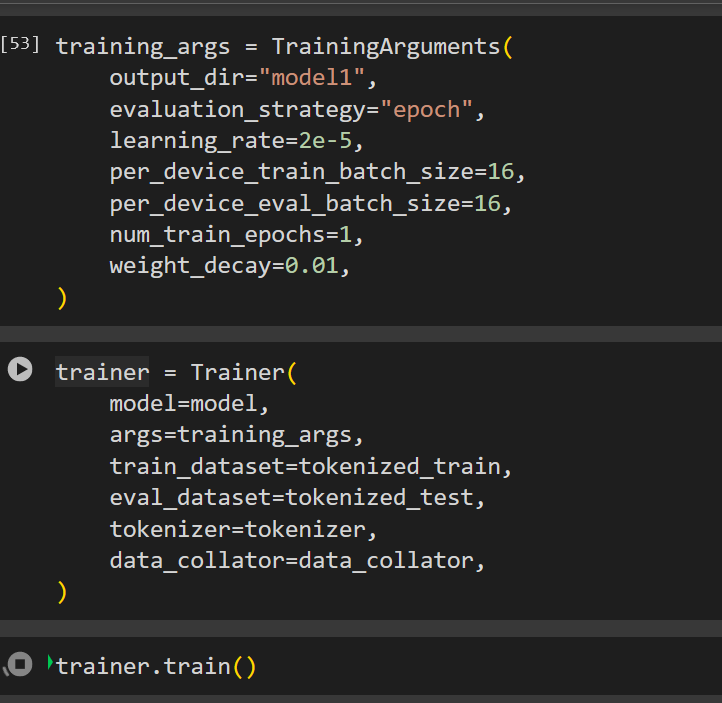


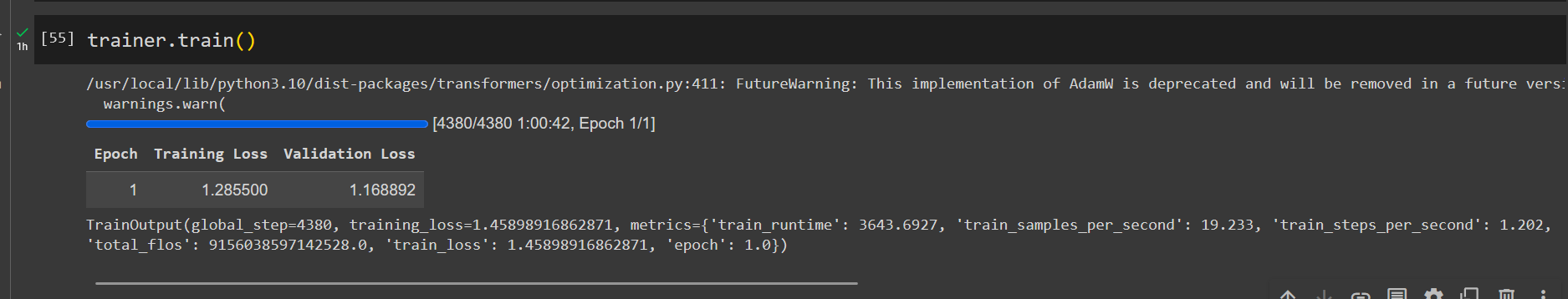
Applying it to the dataset:

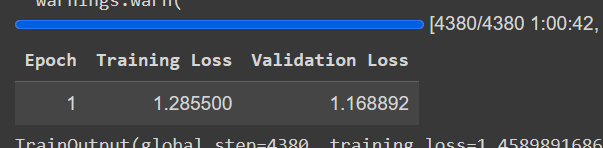


Defining training arguments, and hyperparameters, and training pretrained distilBERT model using pytorch Trainer:









**APPLICATION:**

A web application was created to provide an interface to our model.

