**Text Summarization using Transformers**

GROUP 20

Harani Arumalla-11658201

Hari Sai Rakesh Yalamanchili-11607171

Venkata Mani Sahith Srungarapu-11665562

**Introduction:**

Text Summarization is a technique in natural language processing aims to concise the extensive text documents into shorter versions and capturing the essential information without losing the context or relevance of the whole text. Among the various approaches to the text summarization, extractive text summarization stands out for the unique method of distilling information unlike abstract to summarization which paraphrases and condensed the idea into new expressions, extractive summarization focuses on identifying and extracting the most critical sentences directly from the source test , In sharing that the output romance still relevant to the provided information.

In this project we are going to use transformer-based models , which Revolutionized the natural language processing in a recent times. Transformers known for their exceptional ability to handle sequential data mostly text and transformers like BERT by google Offers significant advantages in understanding context relevance and importance of different parts of a text. So under the hood the Transformers are using methodology called self-attention mechanism which helps to understand the relation between words and the placement of the words in a sentence Etcetera.

And applying the transformer-based model for extractive summarization this project not only contributes to the advancements of nlp technologies but also provides the end user with a powerful tool to navigate to that day to day tasks. For example a phd student can use this tool to summarize an academic paper and companies can use for social marketing etc.

**Related Works:**

Text summarization is a crucial task in the natural language processing , this task was performed using various methods from traditional algorithms to modern deep learning methods.

Heuristic-based Methods - Early summarization systems often relied on fixed linguistic methods such as extracting percentage from each paragraph , Keyword identification or cue phases. These methods are straightforward but lack ability to understand the context of the sentence.

Moving from the straightforward technique statistical approaches utilized frequency of words and phrases to identify the significance of sentence within a text techniques such as Latent Semantic Analysis (LSA) were popular for identifying the underlying topics in text and summarizing based on topic relevance.

As machine learning evolved , Researchers began using surprise learning models that could learn from datasets of documents paired with human generated summaries. Decision trees , Support Vector Machine and Neural Networks were used.

The advancements of deep learning RNN and various types of long Short term memory (LSTMs) Was used to better capture the sentence and sequential nature of text for both extractive and abstractive summarization.

Introduction of attention mechanism has a significant effect on NLP allowing models to focus on different parts of the text as they generate summaries improving the relevancy and coherence of the output.

The development of transformer architecture such as Bert by Google and GPT by OpenAI changed the landscape of NLP, these models use self-attention to process all parts of text parallel, this significantly improving the efficiency and effectiveness of summarization models.

Paper “Text Summarization with Pretrained Encoders" (Liu and Lapata, 2019)” Demonstrated the effectiveness of using pretend encoder like bird for summarization tasks which influence the subsequent research including the development of T5.

**Methods:**

**Model Used –**

The model employed for the research was the T5(text to text transfer transformer) model , specifically the “t5 small”. As the T5 model is flexibility and efficiency in handling various text-based tasks , including summarization. The T5 model operates by converting all text-based language problems into text-to-text format , where the input and output are always string of text.

**Adaptation for Research Problem:**

To adapt the T5 model for the specific task of text summarization , the following steps were taken:

**Data Preprocessing** : The Dataset used was preprocessed to form a suitable format for the model. This involved selecting columns relevant for summarization , handling missing , and splitting the data into training and validation sets.

**Tokenization**: The text data was tokenized using the AutoTokenizer from the hugging face Transformers Library which is compatible with T5 modem this tab converts the text into a format that the model can process including adding special tokens that are necessary for model to understand the start and the end text.

For the BART model , we have used the BART tokenizer is a critical component of the BART model architecture , responsible for converting input text into a format that is interpretable by the model. It is built on the byte pair Encoding (BPE) algorithm , which allows for efficient text encoding while handling a wide range of characters and word fragments.

The two main classes defined for data handling are “Newsdatasets” and “NewsDataLoader”. There are integral for preparing and supplying data to the model in a way that align with the requirements of the T5 transformer model.

**NewDataset Class:**

Initializing Dataset Variables: It stores the stores source texts and target texts and tokenizer model , and information like max length for source and target texts.

The \_\_getitem\_\_ method process individual data points. The text cleaned by removing the unnecessary spaces and newlines , which could affect the model performance if those are not removed as white space and new lines are also conceded as Chara. By the model.

Tokenization and Encoding: This step convert text into a format that T5 model can process that is tokens. While tokenization we tokenize all the text with same length , so get the same length we used padding.

Then we have generated attention mask that the model uses to get the information on which words should be focused.

In the NewsDataLoader Class , we have performed two task that is

**Train – validation Split** – The setup method divides the data into training and validation sets based on the specified split size. This split is critical for validation the model performance during training and adjusting parameters without overfitting

**Dataloader** - It creates PyTorch dataloaders for both training and validation datasets. These dataloaders handle batching, shuffling, and parallel loading using workers, making the training process efficient.

**Fine Tuning:** The T5 model was fine-tuned on summarization task. This involves training the model on specific dataset in this case news articles so that the model learns the specifics of the suppression task such as length style and content of summaries typical from this data set.

**Initialization (\_\_init\_\_):**

‘model’ and ‘tokenizer’: These are passed to the class and stored as instance variables. The model is typically an instance of a pre trained model T5 model that will be fine tuned.

Train\_step\_outputs and validation\_step\_outputs: Lists store outputs from each training and validation step for analysis and logging.

**Forward Pass (forward):**

This method defines that forward pass of the model. It takes ‘input data’ , ‘attention\_mask’ as inputs. The model computes the outputs based on the provided input and masks. If ‘labels’ are provided it return the loss , which is essential for training as it quantifies how far the model’s prediction are from the actual targets.

**Step Function (‘\_step’):**

Centralizes the computation of the loss for single batch. This function is used during both training and validation steps.

It extracts ‘input\_ids’ and ‘attention\_masks’ from the batch , feeds them to the model and computes the loss based on the comparison of prediction to actual labels.

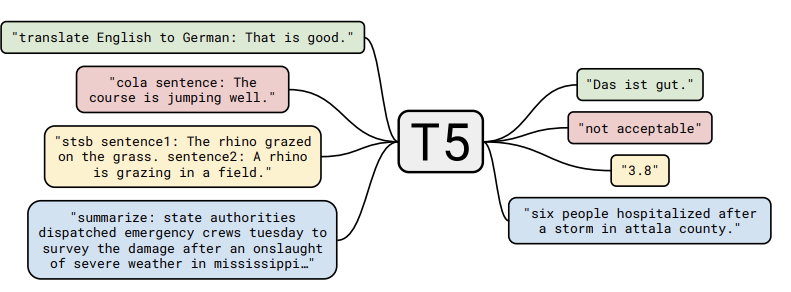
**Training Step (training\_step):**

Utilizes the step function to compute the loss and then store it in train\_step\_outputs for later Averaging logging. This matter returns a dictionary with loss with point of sleeping uses for backpropagation and optimizer steps. Validation Step function is a similar this function.

**Theoretical Underpinnings:**

**T5 model-**

If a model is based on transformer architecture which relies heavily on self attention mechanism. Model to weigh the importance of different words related to each other emotions irrespective of their position. For summary this feature is particularly useful as it is helps the model to focus on most informative parts for the text.



The T5 model are touched to text transfer transformer , revolutionizes Natural language processing tasks by adapting a unified framework where every text-based task is converted into a text to text problem. This means that tasks traditionally requiring specified model architecture such as translations summarizations or question and answering are now treated uniformly on Inputs and outputs are processed as sequels of text. Developed by Google 30 , T5 Model leverages a modified transformer architecture which was originally used in models like BERT and GPT but reimagines its application by emphasizing the generative capabilities of transformers across a variety of tasks.

The fundamental architecture of the T5 consists of encoder and a decoder both of which are based on the transformer model. The encoder processes Input text by converting it into embeddings which are then passed through a series of self attention layers. These layers help the model to weight the importance of different words relating to each other. The decoder on the other hand uses both self attention layer and encoder decoder attention to generate output text from the encode representations.

The process of training D5 involves a pre training phase where the model learns general language understanding from a corpus by predicting masked words similar to BERT. The subsequent fine tuning adjusts the model parameters to specific tasks by training on task specific datasets.

Mathematically the self-attention mechanism at the core of T 5 can be described by the equation ,

Attention(𝑄,𝐾,𝑉)=softmax(𝑄𝐾𝑇𝑑𝑘)𝑉Attention(Q,K,V)=softmax(dk​​QKT​)V ,

Here 𝑄Q, 𝐾K, and 𝑉V Represent the query key and value matrices Derived from the input embeddings and 𝑑𝑘dk is The dimensionality of the keys and queries which scales the dot product to ensure stability gradients. T5 Introduces a unique approach to training by adopting an teacher focusing technique during the fine tuning phase where the model is trained to predict the next token in the sequence given the previous tokens conditioned on the task specific input. This method not only speeds up the Training process but also improves the contextual relevance of the model output making T5 highly effective for a broader spectrum of larger tasks.

**BartForConditionalGeneration –**

BART Is built on transformer architecture which uses self attention mechanism to process input data unlike the original transformer model which uses an encoder for processing the input and decoder for output generation , BART pre trains a sequence -to-sequence model by corruption text With an arbitrary noisy formation and learning to reconstruct the original text.

BART is Pre trained using denoising Auto encoder object. It employs noising strategy such as token masking , text infilling , sentence permutation and document rotation. This pre-training help the model understand context and sequence in text , which is crucial for generation tasks.

The encoder in BART process input text in a bidirectional manner , meaning it considers both left and right context when encoding each token. This results in a richer representation of the input data.

The decoder is BART is auto regressive , meaning it generates one token at a time and uses to previous outputs as additional input geneat6ing the next token.

**Key Equations**

**Self-Attention:**

Attention(𝑄,𝐾,𝑉)=softmax(𝑄𝐾𝑇𝑑𝑘)𝑉Attention(Q,K,V)=softmax(dk​​QKT​)V

Where 𝑄Q, 𝐾K, and 𝑉V are the query, key, and value matrices derived from the input, and 𝑑𝑘dk​ is the dimensionality of the keys.

**Positional Encoding:**

Since transformers do the inherently consider the order of token , positional encodings are added to given the model some information about the relative position of the token in the sequence:

PE(pos,2i)=sin(pos/100002i/dmodel​)

PE(𝑝𝑜𝑠,2𝑖+1)=cos⁡(𝑝𝑜𝑠/100002𝑖/𝑑model)PE(pos,2i+1)=cos(pos/100002i/dmodel​)

**Results :**

**Dataset Description:**

The data set used for this project is available on Cagle under the name of “New summary”. That consists of a collection of new articles along with their summaries making it particularly suitable for summarization tasks.

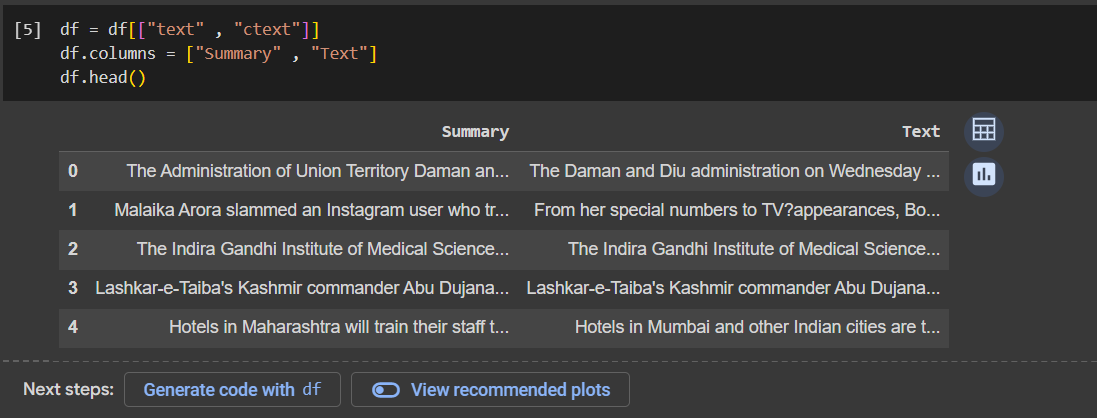
Size : The data set includes several thousand entries that is around 4514 rows.

Features: Each entry typically contains fields for the news article and its corresponding summary.

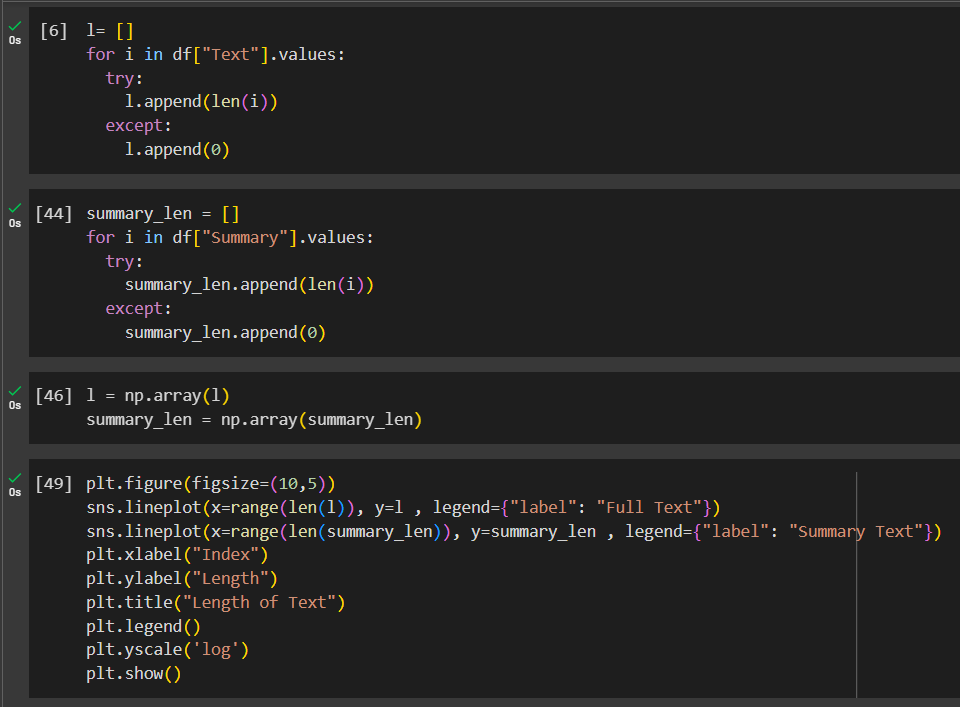
**Initial Data Exploration:**

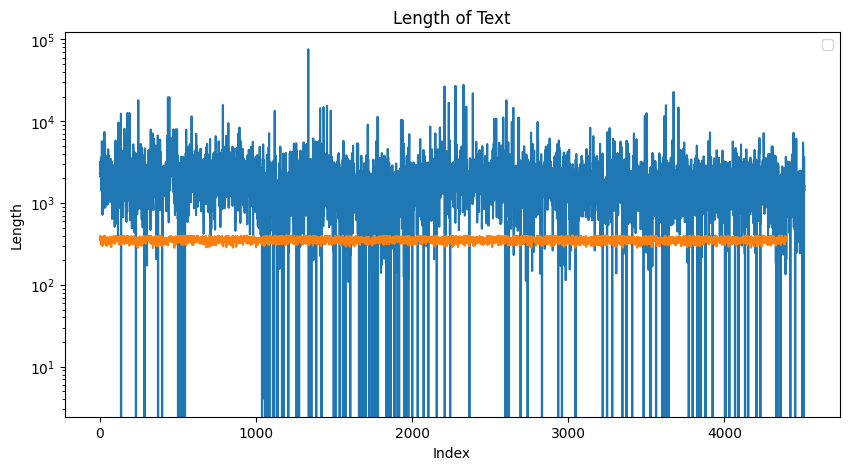
An initial analysis was conducted to understand the distribution of text length. The average length of the full article was found to be greater than summaries within some of the articles reaching out to be 2000 characters while the summaries were generally around 350 characters this analysis was critical for determining the appropriate model input size and for establishing a baseline for compression ratio.

Checking the columns that we will be working on and changing the column

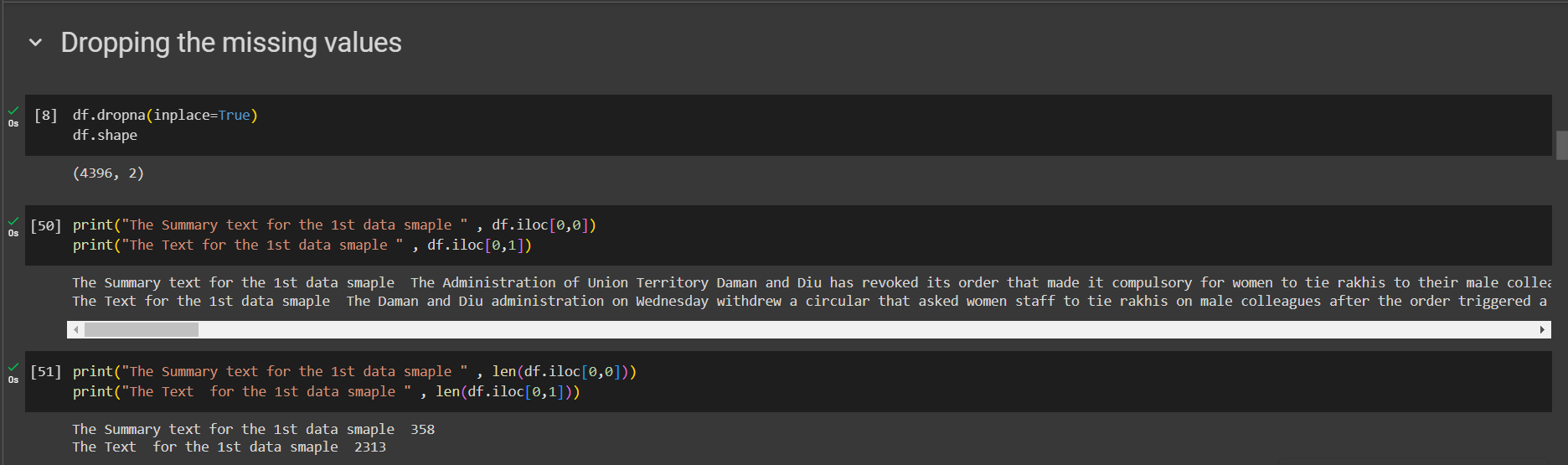


Compared the full article length and summary length:





Removed the missing values and checked the 1st rows in the dataset:

****

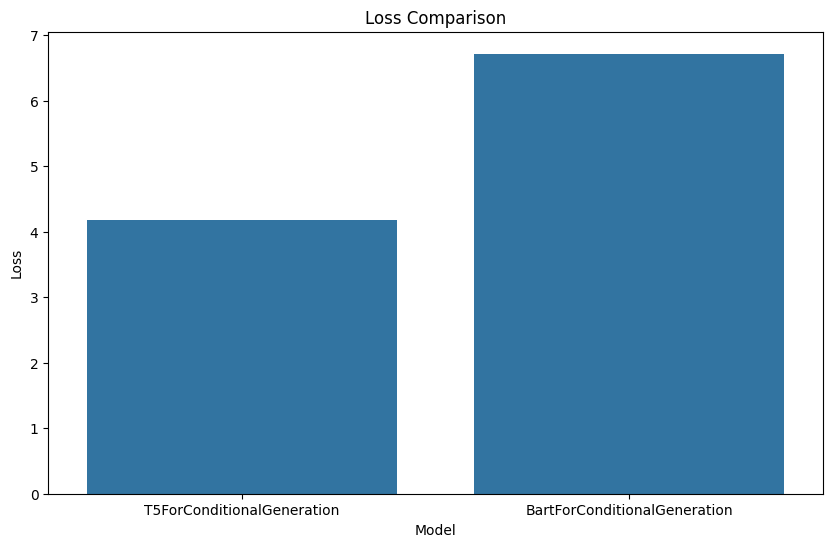
**Model Experiments:**

Two models were chosen for the summarization task, T5ForConditionalGeneration (T5) and BartForConditionalGeneration (BART) . both models are State of the art transformer based architectures with T5 being trained initially to perform a variety of Text to text tasks and BART Being a denoising autoencoder specifically designed for sequence to sequence tasks.

The models for trained under identical conditions to ensure a fair comparison. Each model was fine tuned on our specific dataset for a maximum of five epochs using Adam Optimizer with a learning rate of 2e-5 And a bad size of 4, A ReduceROnPlateau Schedule was amused to adjust the running rate based on validation laws reducing the learning rate by a factor of 0.1 if there was no improvement for three consecutive epochs.

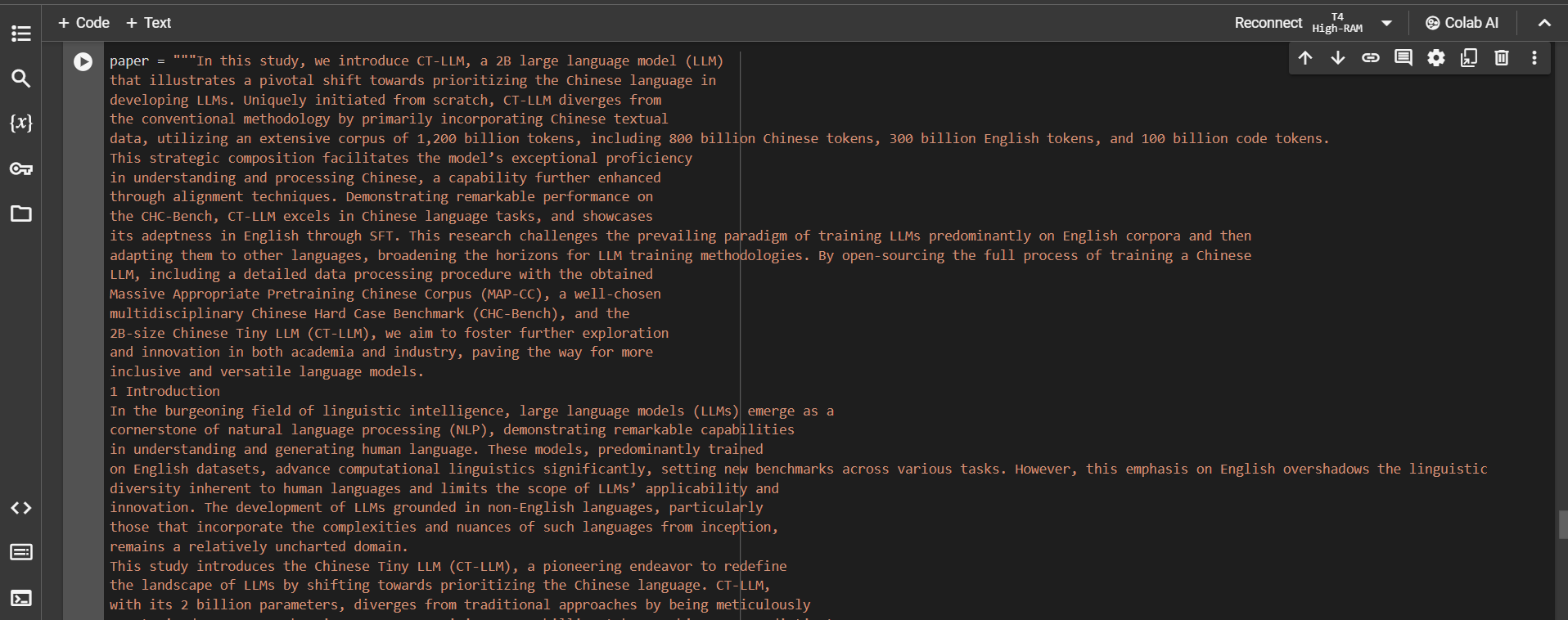
The performance of each model was assessed using validation loss as the primary metric. The results were as follows:

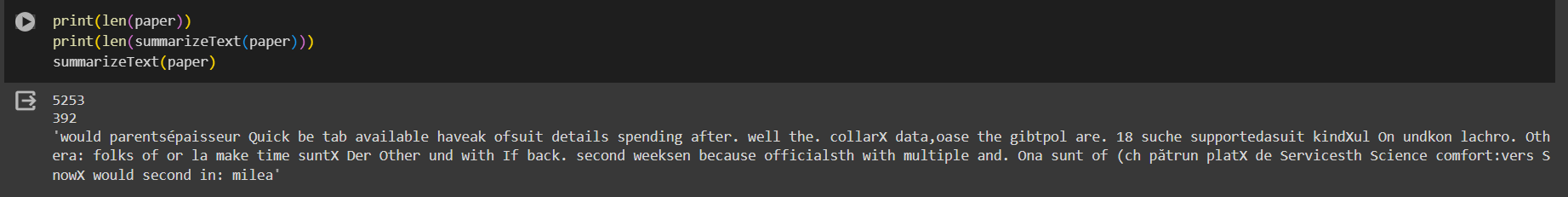
|  |  |
| --- | --- |
| Model | Validation Loss |
| T5ForConditionalGeneration | 4.17 |
| BartForConditionalGeneration | 6.71 |



**Testing the T5 model to summarize a new research paper**







**Discussion:**

**Model performance:**

The T5 model outperformed the BART model , As we can see the above image the validation loss for the T5 model is less than BART model. This performance could be attributed to T 5 text to text pretraining which might be more aligned with the summarization tasks compared to BART’s denoising pretraining objective.

**Shortcomings:**

Despite T’s lower loss this metric alone does not capture the quality to aspect of summarization such as coherence , salience and fluency. Furthermore, both models were limited by the data sets size potentially affecting the generalizability.

**Contextualization with Existing Research:**

The findings aligned with the existing research suggesting that T5 flexible pretraining is effective across various NLP tasks including summarization. However, as noted in other research , BART’s Performance could potentially improve with domain specific fine tuning which was not explored.

**Future Research Perspective:**

* Extended fine tuning with large and more diverse data set to enhance model robustness.
* Incorporation of additional evolution metrics such as ROUGE to provide a multi-faced assessment of summary quality
* Investigation into few short learning and zero short learning capabilities of these models for summarization without extensive fine tune.
* Comparative analysis of large variants of these models that is T5-large , BART-large to assess the impact of model size on summarization performance.

**Refences:**

Text Summarization with Pretrained Encoders (<https://arxiv.org/abs/1908.08345>)

Survey on Automatic Text Summarization and Transformer Models Applicability (<https://dl.acm.org/doi/abs/10.1145/3437802.3437832>)

A Deep Learning Approach to Extractive Text Summarization Using Knowledge Graph and Language Model (<https://www.proquest.com/docview/2832799261?pq-origsite=gscholar&fromopenview=true&sourcetype=Dissertations%20&%20Theses>)