Thought process

Choosing BigBird Pegasus over t5 or bert for the below reason.

> **BigBird relies on block sparse attention instead of normal attention (BERT's attention) and can handle sequences up to a length of 4096 at a much lower compute cost compared to BERT. It has achieved SOTA on various tasks involving very long sequences such as long documents summarization, question-answering with long contexts.**

There are multiple types of models could use -
Seq2Seq, extractive summarization , transformer based but for bbc news summary, Seq2Seq is good enough.

Process was to get a miniature model working first.

Steps thought:-

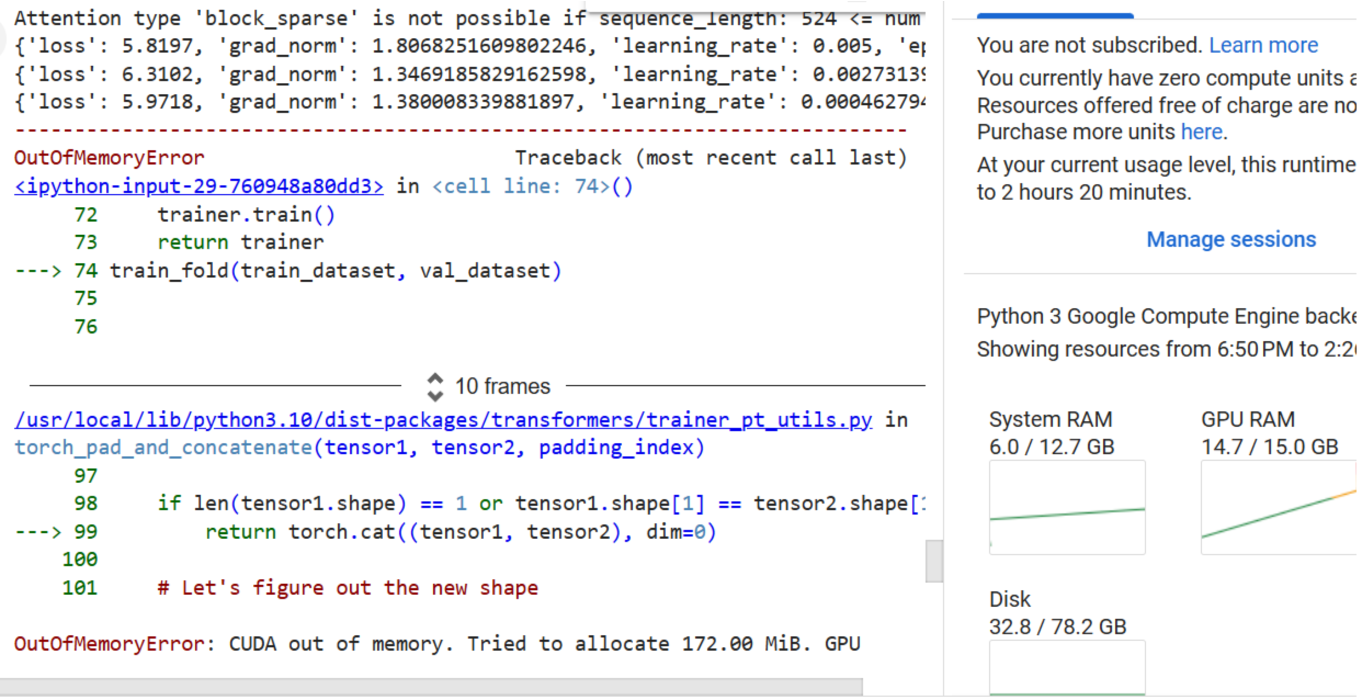
1. Integrating selfmem RAG with bigbird for pre-training
2. Implement InstructDS with BBC news.
3. MT5 vs T5 vs Bart vs Pegasus vs Bert comparison.
4. Use rouge or bleu for evaluation

Challenges Faced:-
CUDA issues, Lots of errors: GPU not assigned by Colab or keep crashing.

```
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xa3 in position 257: invalid start byte
```

```
File /opt/conda/lib/python3.10/site-packages/matplotlib/backends/backend_agg.py:84, in RendererAgg.init(self, width, height, dpi)
82 self.width = width
83 self.height = height
---> 84 self._renderer = _RendererAgg(int(width), int(height), dpi)
85 self._filter_renderers = []
87 self._update_methods()
```

```
ValueError: Image size of 85803x17990 pixels is too large. It must be less than 2^16 in each direction.
```



```
Attention type 'block_sparse' is not possible if sequence_length: 524 <= num
{'loss': 5.8197, 'grad_norm': 1.806825160980246, 'learning_rate': 0.005, 'e
{'loss': 6.3102, 'grad_norm': 1.346918582916259, 'learning_rate': 0.0027313
{'loss': 5.9718, 'grad_norm': 1.380008339881897, 'learning_rate': 0.00046279
---------------------------------------------------------------------------
OutOfMemoryError                          Traceback (most recent call last)
<ipython-input-29-760948a80dd3> in <cell line: 74>()
     72     trainer.train()
     73     return trainer
---> 74 train_fold(train_dataset, val_dataset)
     75 
     76 

                              10 frames

/usr/local/lib/python3.10/dist-packages/transformers/trainer_pt_utils.py in
torch_pad_and_concatenate(tensor1, tensor2, padding_index)
     97 
     98     if len(tensor1.shape) == 1 or tensor1.shape[1] == tensor2.shape[
---> 99         return torch.cat((tensor1, tensor2), dim=0)
    100 
    101     # Let's figure out the new shape

OutOfMemoryError: CUDA out of memory. Tried to allocate 172.00 MiB. GPU
```

You are not subscribed. Learn more

You currently have zero compute units a
Resources offered free of charge are no
Purchase more units here.

At your current usage level, this runtime
to 2 hours 20 minutes.

Manage sessions

Python 3 Google Compute Engine backe
Showing resources from 6:50 PM to 2:2

System RAM
6.0 / 12.7 GB

GPU RAM
14.7 / 15.0 GB

Disk
32.8 / 78.2 GB