# CS521\_Biomedical\_Text\_Summerization

## Dataset

The SumPubMed medical summarization corpus [Gupta et al., 2021] served as the foundation for our model training and evaluation.

Cloning the dataset -!git clone https://github.com/vgupta123/sumpubmed

## Knowledge Graph Creation

Our overall goal was to identify and represent connections between medical concepts extracted from text, in order to enrich transformer-based language models with a graph-based contextual backing of medical relationships. As our corpus for analysis, we utilized 5000 medical document summaries from the SumPubMed dataset.

Our pipeline began by parsing the SumPubMed documents using SciSpacy, a SpaCy model optimized for scientific and medical text. SciSpacy identified critical medical entities and relationships within each document. We then constructed medical knowledge graphs capturing connections between extracted concepts using the Textacy and NetworkX Python libraries for graph analysis. In these graphs, nodes represented medical concepts while edges linked related concepts based on co-occurrence within documents or explicit connections made in the source text.

## Model selection

Issues with GPT-2 and BART -

Very less context limit of 1k tokens

Issues with T5 -

Model too large for our hardware constraints

Model chosen to move forward with - Longformer Encoder-Decoder - allenai/led-base-16384

Advantages -

- Able to finetune it on M3 Max Macbook

- has high token limit of 16k tokens

## Implementation

We carried out rigorous experiments, implemented in PyTorch using the Transformers library which provides state-of-the-art models optimized for sequence tasks. PyTorch enabled rapid and modular prototyping of graph incorporation approaches. The TensorBoard toolkit was leveraged for tracking experimental results across metrics like ROUGE and loss convergence over multiple runs.

For knowledge graph creation - Our computing infrastructure consisted of Google Colab notebooks accessed through the cloud, which provide free GPU access. This enabled executing jobs in a fast, parallelized manner critical for neural approaches. All models were trained using T4 GPUs. Reason - SciSpacy dependency issues on local environment.

For model finetuning - We used M3 Max Macbook pro with 36 GB RAM.

## Evaluation -

We used ROGUE metrics for evalluating the 2 models with 100 randomly picked text-summary paires from the sumpubnet dataset-

1. Finetuned with sumpubnet data

2. Finetuned with Sumpubnet data along with Knowledge graph embeddings

## Results -

We noticed an increase in precision and f1 scores for the model with knowledge graph embeddings.

## Future Work -

We would like to continue working on the project and explore other techniques of creating and embedding knowledge graphs to improvwe the results further.