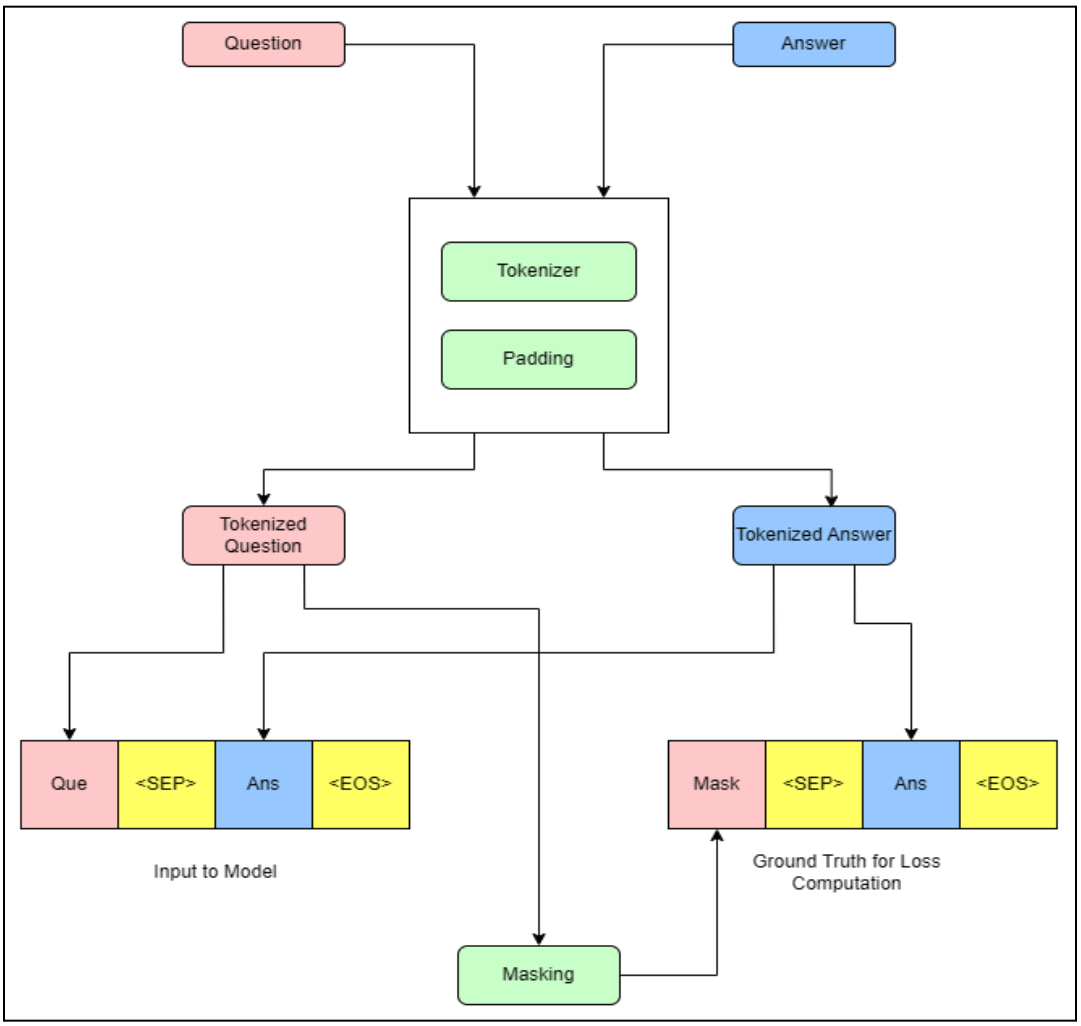
Background: Question Answering (QA) in NLP involves training models to understand questions posed in natural language and extract relevant information from a given context or dataset to provide accurate and concise answers. I'm tasked with creating a question-answering system for Medical QA to understand and respond to natural language questions using the **MedQuad-MedicalQnADataset** in Natural Language Processing (NLP).

Dataset: MedQuad-MedicalQnADataset is a specialized dataset tailored for the field of medicine and healthcare. It's designed to facilitate the development and training of Question Answering systems within the medical domain. This dataset comprises a collection of questions and corresponding answers formulated in natural language, specifically curated to cover diverse medical topics and scenarios. Its focus on medical queries makes it a valuable resource for NLP researchers and practitioners aiming to build and enhance QA systems dedicated to healthcare-related inquiries.

Architecture: To accomplish this task, the GPT2 model is fine-tuned over the given medical dataset. Since GPT2 is a decoder-only model, each sample of the question-answer pair is concatenated with [SEP] token in between and then passed to the GPT2 model.



For eg.

Question- What are marine toxins?

Answer - Marine toxins are naturally occurring chemicals that can contaminate certain seafood.

Input to GPT2 model -

Each sentence is fed to the GPT model in the given format:

<question> [SEP] <answer>

Example:

*What are marine toxins? [**SEP**] Marine toxins are naturally occurring chemicals that can contaminate certain seafood.*

Similarly, the ground truth is modified into the following format:

<mask> [SEP] <answer>

Example:

<MASK> <MASK> <MASK> <MASK> <MASK> [***SEP***] Marine toxins are naturally occurring chemicals that can contaminate certain seafood.

<MASK> is one of the special tokens added to the GPT2 tokenizer to MASK the question in the ground label so that while generation, softmax will nullify those parts of input and only focus on generating answers.

To make the input length the same for all training inputs, the desired number of paddings before the Question and after the answer were added.

Original data is divided into train, test and validation sets with approx 7400 in train data, 5400 in val data and 3200 approx. in test data. This division is done using the test_train split function of Sklearn.

Various checkpoints are saved at regular intervals during training. Checkpoints are then evaluated by validation data perplexity and the checkpoint with the lowest perplexity is then considered as final model to be used for test data.

For final inference, only the question with [SEP] token appended is passed to the model. Auto regressive technique is used to generate answer until <EOS> token is achieved or some fixed length token are generated, whichever comes first.

For evaluation, rouge score can be used for the final generated answers and the golden answers.

Challenges -

GPT2 model is used which is trained on a very large dataset. It was fine-tuned on only 7400 question-answer samples. Since the training dataset is very small, it is not able to generate very good answers.

Due to resource constraints, the model was being trained only for 3 epochs and with GPT2 small. Performance can be improved with large training samples and GPT2 large trained on more number of epochs.

Another approach that can also be used -

Each Question sample can be encoded by using a sentence transformer to a fixed dimension vector. For every new test sample, encode it using a sentence transformer and find the nearest question that matches the test question and return the answer of that question.

The training dataset and models are stored in Google Drive. Link