EAI Endorsed Transactions
on AI and Robotics
Research Article EAI.EU

Implementation of GPT models for Text Generation in Healthcare Domain

Anirban Karak[1], Kaustuv Kunal[2], Narayana Darapaneni[3], Anwesh P R[2,*]

[1] PES University, Bangalore, Karnataka, 560050, India
[2]Great Learning, Hyderabad, Telangana, 500089, India
[3]Northwestern University, Evanston, IL 60208, United States

Abstract

INTRODUCTION: This paper highlights the potential of using generalized language models to extract structured texts from natural language descriptions of workflows in various industries like healthcare domain
OBJECTIVES: Despite the criticality of these workflows to the business, they are often not fully automated or formally specified. Instead, employees may rely on natural language documents to describe the procedures. Text generation methods offer a way to extract structured plans from these natural language documents, which can then be used by an automated system.
METHODS: This paper explores the effectiveness of using generalized language models, such as GPT-2, to perform text generation directly from these texts
RESULTS: These models have already shown success in multiple text generation tasks, and the paper's initial results suggest that they could also be effective in text generation in healthcare domain. In fact, the paper demonstrates that GPT-2 can generate comparable results to many current text generation methods.
CONCLUSION: This suggests that generalized language models can increase the efficiency and accuracy in text generation, where workflows are repetitive and sequential.

Keywords: healthcare, text generation, GPT-2, PubMed dataset, medicine, NLP

Received on 05 October 2023, accepted on 05 April 2024, published on 09 April 2024

doi: 10.4108/airo.4082

*Corresponding author. Email: anwesh@greatlearning.in

1. Introduction

Lifestyle diseases are continuously on the rise, due to the sedentary lifestyle of human beings, in today's world. Research and development in the medical community is progressing at a rapid rate but the number of doctors and health professionals produced is less compared to the rise in the world's population. They can only spend a limited amount of time diagnosing the patients and are often unable to answer all the queries of the patients due to lack of time. The patients then reach out to the healthcare professionals who would benefit from having the information related to the usage of the medicine handy, to assist the patients. On knowing exactly why the medicine was prescribed, the benefits as well as the side effects of these medicines, the patients too will have a better chance of recovery.

Out of 8 billion people, 400 million people visit hospitals to get health checkups every year. The next year when they revisit again, the healthcare professional will want to see their previous year's diagnosis. Doctors have limited time to cater to these huge population, and it will be impossible for an individual to tally the reports of health checkup of previous years (along with other issues like major operation- an individual might have undergone in previous years) in a short time, and this also has a possibility of misinterpretation of report leading to error. Thus, having the key information regarding the medicine prescribed in the last year along with the usage, will greatly help in time saving.

Text generation involves creating new text based on a given input. This technology uses advanced algorithms to generate text that is often indistinguishable from human-written content. As a type of artificial intelligence (AI), text generation can produce coherent and grammatically correct sentences that are comparable to what a human would write.

The applications of text generation are varied and diverse [1]. For example, it can be used for summarization tasks, where it generates concise and meaningful summaries of lengthy documents or articles. It can also be utilized in dialogue systems to create engaging and interactive conversations with users. Furthermore, text generation is an essential component of machine translation, where it helps to convert written text from one language to another accurately. Overall, text generation is a powerful and innovative technology that is changing the way we interact with language.

Healthcare professionals sometimes face challenges to provide accurate information to patients, in the absence of a trained professional like doctors. Remembering the usage of all the prescribed medicines along with their benefits and side effects is a challenging task. GPT-2 might be useful in these scenarios, as it has proven to be useful in abstractive text generation. In this paper, we explored the possibility that when medicine name is passed to GPT-2's prompt, it will be able to summarize and generate text-based output regarding detailed description of the usage of the medicine. This will resolve manual retrieval challenges as well as the accuracy of the model will not be compromised. Incorrect diagnosis or delay in treatment will be overcome by GPT-2's text generation.

To overcome this issue, in this paper, we tried to explore the possibility that given a medicine name, the ability of GPT-2 to correctly interpret the data and provide answers regarding the usage of this medicine. Through investigations, we have discovered that these models are able to achieve comparable, and in some cases superior scores, in comparison to previous methods.

To test the efficacy of GPT-2 in this context, we have analyzed natural language texts from PubMed dataset and evaluated the model's performance based on its Rouge score - a widely used quantitative measure for this task. Furthermore, we have compared the results with those previously published for text generation models, which utilize a range of techniques such as reinforcement learning and deep learning. The findings suggest that GPT-2 holds immense promise as a powerful tool for generating structured texts from natural language text prompts.

In this paper we have taken the dataset called PubMed from National Library of Medicine (NLM) that provides access to biomedical literature, including journal articles and book chapters. PubMed is a crucial resource for healthcare professionals, researchers, and academics in the biomedical field. The PubMed dataset is a collection of metadata and abstracts of biomedical literature that are indexed in PubMed. The dataset contains structured information about articles, including the title, authors, abstract, publication date, journal name, and more. The PubMed dataset is available for free and can be accessed through NCBI website. Additionally, the dataset is available for download in various formats, including XML, CSV, and JSON. Researchers and data scientists can use the PubMed dataset to perform various types of analysis.

2. Related Work

Established in 2015 as a non-profit organization, OpenAI initiated the development of GPT as part of its research agenda to promote and create "friendly AI" that would serve the betterment of humanity. The model was first introduced in 2018, containing 117 million parameters. The following year, OpenAI released an even more sophisticated version, GPT-2, featuring 1.5 billion parameters. In comparison, GPT-3, the latest version, boasts a massive 175 billion parameters, surpassing its predecessor by more than 100 times and outperforming other similar programs by ten times [2].

Earlier models, including BERT, had demonstrated the potential of the text generator approach, showcasing the remarkable capabilities of neural networks in producing extensive pieces of text, which were previously deemed impossible. To prevent any potential issues, OpenAI cautiously released access to the model in increments to observe its usage [3]. During the beta phase, users had to apply to use the model, and access was initially free of charge. However, the beta phase ended in October 2020.

Partnership with OpenAI provides Microsoft with the opportunity to integrate GPT-3 into its own products and services. In November 2022, ChatGPT was launched, and it was initially free for public use during its research phase. This launch brought GPT-3 more mainstream attention, providing many non-technical users with the opportunity to try the technology.

3. Materials and Methods

3.1. Dataset

PubMed is a free search engine maintained by the National Library of Medicine (NLM) that provides access to biomedical literature, including journal articles and book chapters. It is a crucial resource for healthcare professionals, researchers, and academics in the biomedical field. The PubMed dataset is a collection of metadata and abstracts of biomedical literature that are indexed in PubMed. The dataset contains structured information about articles, including the title, authors, abstract, publication date, journal name, and more. The PubMed dataset is available for free and can be accessed through NCBI website.

3.2. Model Architecture

The use of transformer-based architectures[4] has become widespread in newer language processing models, as they leverage attention mechanisms for converting input sequences to context preserved output texts. The model was mostly trained on five important tasks – data collection was done from PubMed site via the PubMed Fetcher library, collating the tokenized NER data and labels, fine tuning the token classifier, preprocessing the raw text for GPT-2 and the GPT-2 model was trained for text generation.

3.3. Training strategy of the model

First, we load GPT2DoubleHeadsModel and GPT2Tokenizer for training. Training consists of two task train and evaluate. Before training special token ('bos_token': '<|startoftext|>','eos_token':'<|endoftext|>','pad_token':'<pad> ','additional_special_tokens':'<|keyword|>', '<|summarize|>']) were added to the GPT2 tokenizer and resized the token embeddings. Training and validation data were loaded using pytorch data loader. The train method accepts five parameters which are batch (tensor data), number of iterations, GPT2 model, optimizer, and scheduler.

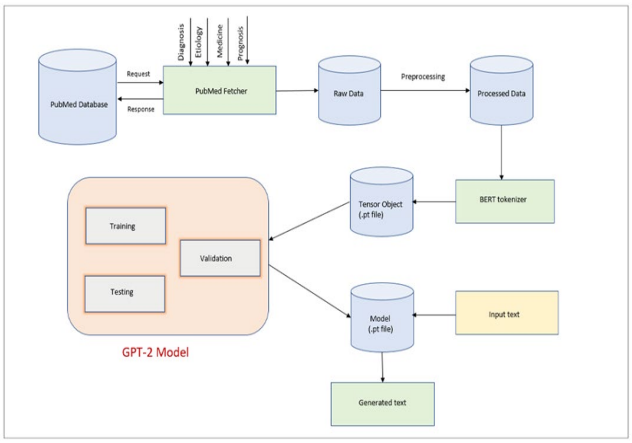

Figure 1: Architecture Diagram of the proposed Model

Adam optimizer was used with lr=5e^{-5} and eps=1e^{-8} for training. We used *get_linear_schedule_with_warmup* scheduler with 50 warm up steps from transformer. During loss calculation, three losses were calculated for the training - LM loss (Language model) [5], MC loss (Multiple choice) and total loss.

$$\textbf{Total loss} = \frac{lm\text{-}loss * lm\text{-}coeff + mc\text{-}loss * mc\text{-}coeff}{accumulating\ gradient} \quad (1)$$

We used clip_grad_norm_ from torch utils package to prevent the exploding gradient problem. Training was configured with 20 epochs. The evaluate method was used for validation. It took two params - validation tensor data and GPT2 model, to perform the validation. After successfully training the model, the training loss logs were saved in the disk for future use.

3.4. Evaluation based on metrics

Rouge Score

It is a software package and a set of metrics that are commonly used to assess the performance of machine translation and automatic summarization software in natural language processing. It evaluates the quality of a machine generated summary by comparing it with human-produced summaries or translations, known as references. The metric utilizes N-gram overlap between the machine-generated and gold summaries to determine the ROUGE score. Specifically, precision in ROUGE calculates the ratio of n-grams that might be recurring in generated as well as gold summary.

Recall

The ROUGE metric's recall score measures the degree of overlap between the n-grams found in the reference and the model output. It is represented mathematically as:

$$\frac{n\ grams\ found\ in\ (model\ +\ reference)}{\text{total}\ n\ grams\ present\ in\ the\ reference}$$

Precision

Precision is computed in a similar manner to recall, except that it divides the overlapping n-grams by total of n-grams in the model output, rather than in the reference summary. It is represented mathematically as:

$$\frac{n\ grams\ found\ in\ (model\ +\ reference)}{total\ n\ grams\ present\ in\ the\ model}$$

F1-Score

F1 score is defined as:

$$2 * \frac{precision * recall}{precision + recall}$$

BLEU Score

BLEU is a commonly used algorithm for evaluating the quality of machine-translated text, which aims to measure similarity of machine-generated text to a human translation, which is considered a benchmark for translation quality. BLEU achieves this by comparing n-grams in the machine translation with those in one or more reference translations and calculating a precision-based score.

BLEU score ranges from 0 to 1, with 1 indicating a perfect match between the machine-generated text and the reference translation. In particular, BLEU may not be able to accurately assess translations that use different words or syntax than those found in the reference translation, even if the overall meaning is preserved.

Overall, the BLEU score provides a quantitative measure of the performance of machine translation systems, which can help to objectively assess their accuracy and effectiveness. By examining the n-gram overlap between machine and human translations, BLEU provides a useful tool for researchers and practitioners alike to measure the quality of machine-generated text. It is represented as:

$$\underbrace{\min\left(1,\ exp\left(1-\frac{\{\text{reference-length}\}}{\{\text{output-length}\}}\right)\right)}_{\text{brevity-penalty}}\ \underbrace{\left(\prod \text{i}=1^{4}\ \{\text{precision i}\}\right)^{1/4}}_{\text{n-gram overlap}}$$

$$precision_i = \frac{\sum_{\text{snt}\in\text{ Cand-Corpus}}\ \sum_{i\in\text{snt}}\ \min\left(m^i_{\text{cand}}, m^i_{ref}\right)}{w^i_t = \sum_{\text{snt}'\in\text{ Cand-Corpus}}\ \sum_{i'\in\text{ snt}}\ , m^{i'}_{\text{cand}}}$$

Where,

- m^i_{cond} - number of i-gram in candidate like translation of reference
- m^i_{ref} - number of i-gram in translation of reference
- w^i_t - number of i gram in translation of candidate

Cross Entropy Loss:

Cross-entropy loss is a popular loss function in natural language processing that is used to train language models. It is used to approximate true probability distribution. It is often combined with other metrics such as perplexity to generate comprehensive evaluation of the model's performance.

Cross-entropy loss is commonly used in many natural languages processing applications, including machine translation, sentiment analysis, and text classification. By minimizing the difference between predicted and true probability distributions, language models can learn to generate more accurate and relevant outputs for variety of tasks. It is represented mathematically as below:

When x is continuous, $\boldsymbol{H(x)} = -\int_x \boldsymbol{p(x)log\ p(x)}$

When x is discrete, $\boldsymbol{H(x)} = -\sum_x \boldsymbol{p(x)\ log\ p(x)}$

Methodology

PubMed is a free search engine developed by the National Library of Medicine (NLM) at the National Institutes of Health (NIH) in the United States. It provides access to a vast collection of biomedical literature, including scientific articles, books, and conference proceedings. We cannot directly download the data. To download the data, we used PubMed Fetcher python library which takes different kinds of key words such as diagnosis, medicine etc. and gives the related data, which was saved in csv format. The raw data was not cleaned, so different kinds of pre-processing such as null removal, duplicate removal, special character removal, emoji removal etc. had been performed to clean the data.

Apart from cleaning, one important step has been performed which is Part-of-Speech tagging. By identifying the part of speech of each word in a sentence, the computer could understand the grammatical stricture of the sentence,
which is essential for accurately interpreting the meaning of the text. POS tagging was performed using python NLTK library. The final step before model training was tokenization. BERT tokenizer was used for tokenization. The BERT tokenizer is based on WordPiece, a sub word tokenization method that breaks down words into characters, which allows BERT to handle out of vocabulary words and capture more fine-grained details of language. Training, Validation, and testing was performed on tokenized data using Distilled GPT2. After the training, model was saved in .pt file format in secondary disk for future use. Now if any input text is passed to the model, it will generate the relevant text.

Table 1: Comparison of output (generated text) with and Without Beam Search

Medicine	Beam Search	Sampling
Bicalutamide	Bicalutamide (Casodex), a non-steroidal anti-androgen, has been shown to be effective in the treatment of patients with advanced prostate cancer. The aim of this study was to evaluate the efficacy of bicalutamide in the treatment of patients with advanced prostate cancer. PATIENTS AND METHODS A total of 30 patients with advanced prostate cancer were treated with bicalutamide (1 5 0 mg /', 'day).RESULTS Of the 3 0 patients, 2 2 (7 3. 3%) showed a decrease in serum prostate-specific antigen (PSA) levels. The median duration of PSA response was 1 2 months. The median time to disease progression was 1 2 months and the median survival time was 1 8 months. The most common adverse events were gynecomastia and hot flushes.	{'generated_text': 'Bicalutamide (1 mg / kg.body wt) was injected subcutaneously once daily for 6 0 days, and rats were anesthetized to obtain blood specimens. Serum testosterone, ALT / AST levels, activities'} {'generated_text': 'Bicalutamide, which has not been associated with the development of gynecomastia, caused breast pain in one patient. One man was lost to follow up after he had had his second dose of bicalutamide because'} {'generated_text': 'Bicalutamide (1 0 μM). This observation was also consistent with the decreased AR nuclear localization in response to Bical'} {'generated_text': 'Bicalutamide treatment in the absence of androgens enhanced AR degradation. Therefore, we determined whether the observed enhanced degradation of AR was due to ubiquitin-mediated protein degradation. Both LNCaP and C4 – 2B cells'}

	CONCLUSION Bicalutamide is an effective and well-tolerated treatment for patients with advanced prostate cancer.	

4. Results

This model has the capability to generate text using beam search and top k sampling. Beam search is a popular heuristic algorithm used in natural language processing (NLP) for finding the most likely sequence of words in a sequence-to-sequence model, such as in neural machine translation or text generation tasks. The size of the beam, k, is a hyperparameter that determines how many sequences to keep track of.

At each time step, the algorithm expands each sequence in the beam by generating all possible next words, and scores each resulting sequence based on the probabilities assigned to the candidate words by the model. The top k sequences with highest scores are kept in the beam, while others are discarded. The algorithm repeats this process for each subsequent time step till end of sequence is reached or maximum length has been covered.

Finally, the sequence with the highest score in the final beam is chosen as the output. Generally, beam search takes more time than top-k sampling, but it generates quality text. As can be seen from table 1, Beam search produced more relevant outcome compared to sampling.

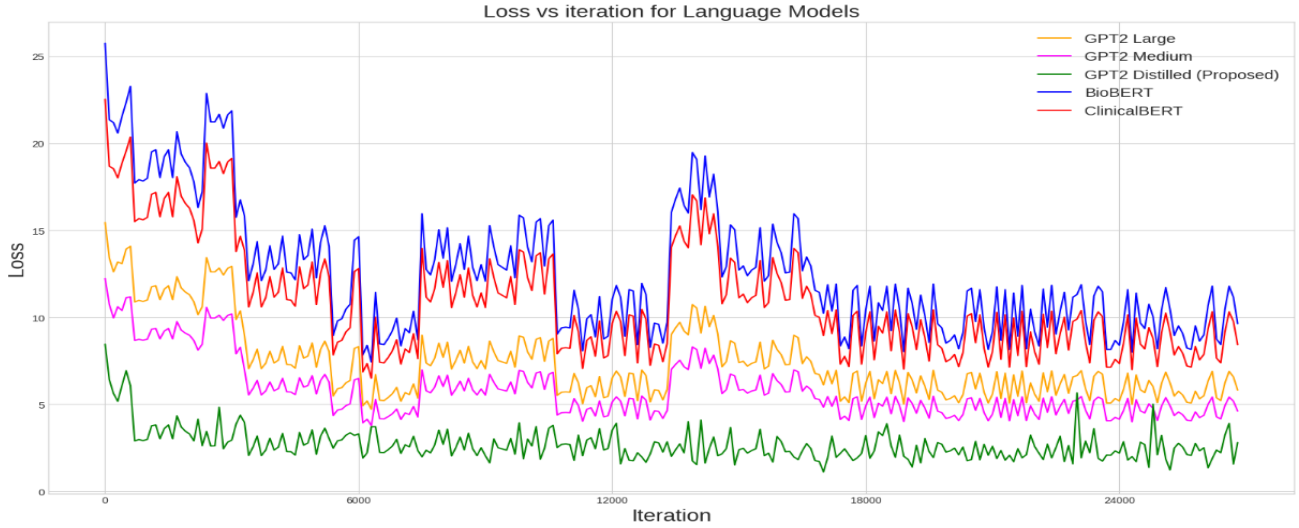

Figure 1: Loss Comparison of the model

As seen in figure 2, we can see that the total loss for BioBERT is highest, it starts from 26 and ends at 10 whereas ClinicalBERT performs better than the BioBERT. The total loss for ClinicalBERT starts at 23 and ends at 9. GPT2 large and GPT2 medium performance is slightly better than the BERT. For GPT-2 larger model loss starts at 15 and ends at 6 and GPT2 medium loss starts at 12 and ends at 4. GPT2 Distilled out performed all the models. The loss starts at starts at 9 and ends at 1. The language model loss is in exponential form. After 600 iterations language model loss drop significantly. The multiple-choice loss drops to zero very quickly. Initially total loss was also high, and it followed the language model loss pattern and then drop significantly after 700 iterations.

Rouge Score evaluation:

We calculated rouge metrics on recall (r), precision (p) and f1-score (f) as shown in table 3. Rouge- 1 score was above 0.45, Rouge -2 score was above 0.56 and Rouge-L score was above 0.76 for all 3 metrices.

We then proceeded to plot bar chart of rouge score on average values.

Table 1: Rouge Score Metric

metric	Rouge Score		
	Rouge-1	Rouge-2	Rouge-3
r	0.450045	0.549370	0.763382
p	0. 458857	0.660528	0.775071
f	0. 446232	0.591211	0.759433

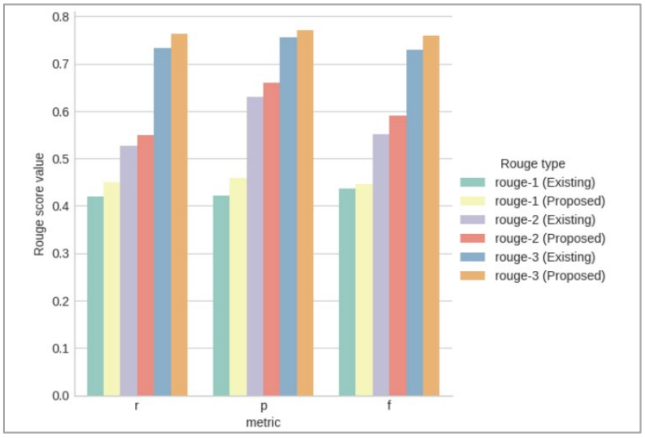

Figure 3: Plot of Rouge score on average value

In evaluating the similarity between generated text, we utilized the precision version of ROUGE. Specifically, ROUGE-n with 0.5 precision indicates that 50 percent of the n-grams might recur in golden summary.

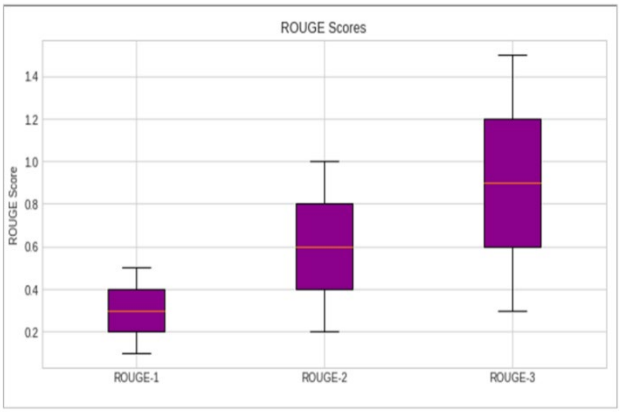

Figure 4: Rouge score on precision value

As seen in figure 4, the result of Rouge-3 has a better score compared to Rouge-2, but Rouge-2 performed better compared to Rouge-1. As Rouge-3 has a median close to 0.9, so it performed better on the longest sequence. This model performs best on the trigrams and has a satisfactory performance on bigrams. We see some outliers on doing box plot but since its negligible, hence we ignore the same. Nonetheless, the ROUGE score of the GPT-2 summarizer was generally high, indicating that it performed well.

BLEU Score evaluation:

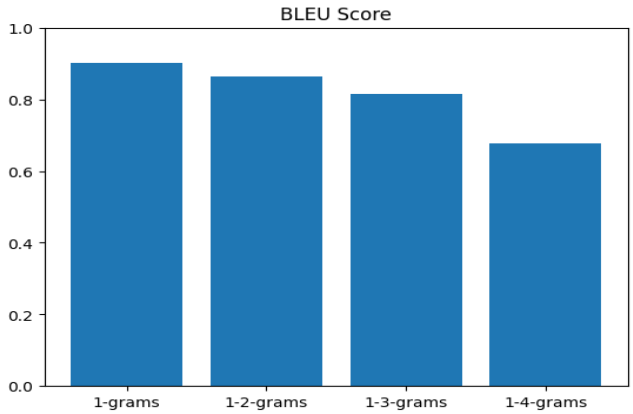

Figure 5: BLEU score of the model

From figure 5, we can see that the model performs the best on 1 gram with a score of 0.9, for 1-2 grams the model generates a score of 0.86, for 1-3 grams the model generated a score of 0.82, for 1-4 grams the BLEU score generated was 0.67. The observation here is that when more context of the word is captured the BLEU score tends to go down but since it generated a score of 0.67, even for 1- 4 grams, hence it can be concluded that the model performance was pretty good.

5. Discussions and Conclusions

Text generation is still a challenging task for deep learning NLP, especially when it comes to domain-specific corpora especially medical domain that differ from the pre-training data. Medical data are not easily available. There are few sites that publish public medical data.

We took the subset of PubMed data in few categories such as Diagnosis, Medicine, Prediction, Prognosis, Sepsis etc. To create the training data set for our text generation model, we merged 5 tensor objects, each containing 4 options in a multiple-choice format. These options corresponded to 4 keyword-summary pairings, with only one of the pairs being correct. The input sequences in the dataset had a shape of [4,1024], while the token type sequence was [4,1024].
Additionally, we included the gold abstract label for the generation task as [4,1024], the last token label as [4], and the multiple-choice answer as [1]. The training dataset for the model being used for multiple choice tasks is composed of five tensor objects, which are multi-dimensional arrays of numerical values. These tensors are integrated to form the training dataset. The shape of input sequences in the tensor is [4,1024], indicating that it comprises 4 sequences of 1024 values each. The token type sequence tensor also has a shape of [4,1024], signifying that it contains 4 sequences of 1024 values each, used to identify the different token types in the input sequences. The tensor object used to store the gold abstract labels for the generation task has a shape of [4,1024], indicating that it comprises 4 sequences of 1024 values each. These labels are used to signify the correct text for each input sequence. The last token label tensor has a shape of [4], it contains 4 values and is used to identify the last token of the

input sequences. The multiple-choice answer tensor has a shape of [1], it contains 1 value and is used to identify the correct answer among the 4 choices. The goal of this model is to predict the correct pair among these 4 choices.

While the model has not yet reached human-level performance, its results remain interpretable. We have measured the performance of our model with different GPT-2 model and BERT model. BioBERT and ClinicalBERT did not performed well as compared to GPT-2 medium and GPT-2 large. The total loss was more for BERT model as compared to GPT-2. Among GPT-2 models, Distilled GPT-2 performed well. Distilled GPT-2 has the lowest loss compared to other models.

One potential method of improving could be to continue training the model using full PubMed data set along with AERS, VAERS and FAERS dataset. If we trained this model on diverse medical data then the quality of generated text will improve, and the model performance could benefit if trained on larger datasets and by using higher GPU.

To run this model on V100 GPU for 30k iterations, it took me around 30 hours, if we can use higher computation power and can run the model on entire dataset, the performance might be faster, and we can weed out those articles that have irrelevant information. we can use GPT-2 XL for training and fine tuning this will require more computing resource.

References

[1] Virapat Kieuvongngam, Bowen Tan, and Yiming Niu. Automatic text summarization of covid-19 medical research articles using bert and gpt-2.arXiv preprint arXiv:2006.01997, 2020

[2] Luo, Renqian, Sun, Liai , Xia, Yingce , Qin, Tao Zhang, Sheng , Poon, Hoifung, Liu and Tie-Yan. BioGPT: generative pre-trained transformer for biomedical text generation and mining. Briefings in Bioinformatics, Oxford Academic, 2022.

[3] Su, Nigel, Yixuan and Collier. Contrastive search is what you need for neural text generation. arXiv preprint arXiv:2210.14140, 2022

[4] Chang, Ernie and Shen, Xiaoyu and Zhu, Dawei and Demberg, Vera and Su, Hui. Neural data-to-text generation with lm-based text augmentation. arXiv preprint arXiv:2102.03556, 2021

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. corr abs/1706.03762 (2017). 2017.

[6] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018. URL-https://d4mucfpksywv.cloudfront.net/better-language models/language-models.pdf

[7] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345, 2019.

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

[9] Zhen Huang, Shiyi Xu, Minghao Hu, Xinyi Wang, Jinyan Qiu, Yongquan Fu, Yuncai Zhao, Yuxing Peng, and Changjian Wang., "Recent trends in deep learning based open-domain textual question answering systems.," 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[11] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing, 2019.

[12] Derek Miller. Leveraging bert for extractive text summarization on lectures, 2019.

[13] Dima Suleiman and Arafat Awajan. Deep learning based abstractive text summarization: approaches, datasets, evaluation measures, and challenges. Mathematical problems in engineering, 2020.