Long Range Dependencies in Biomedicine

Nagacharan Vemula
Arizona State University

I. INTRODUCTION

Recently, the medical domain has witnessed a significant surge in machine-readable text, especially within electronic health records (EHRs), resulting in the development of pre-trained medical language models (LLMs). While numerous LLMs have been tailored for shorter medical texts, the demand for models capable of efficiently processing longer and more complex documents, including comprehensive patient summaries and extensive EHRs, is increasing in real-world healthcare scenarios. A crucial and relatively unexplored challenge lies in ensuring the proficiency of these models in handling extended sequences while capturing essential long-range dependencies.

The constrained input token length window of LLMs poses a bottleneck when confronted with extensive contexts such as Electronic Health Records (EHRs) in medical data. This project is committed to tackling the challenge of capturing prolonged contexts and long-range dependencies in biomedical data. It specifically addresses the constraint imposed by the fixed token length of LLMs, ultimately enhancing their overall performance.

The Project is divided into two stages:

Stage1
During this stage, we conducted an analysis of the data and observed that the maximum token length values in the input data surpass the permissible limit for the model's input. In response, we have opted to truncate the input data to align with the maximum input token length of the model. Subsequently, we evaluated the performance of LLM (In-BoXBART [1]) on several benchmark bio medical datasets (Smoking Challenge Data 2006 [2], Obesity Challenge Data 2008 [3], Heart Disease 2014 Data) using this approach, considering it as a method to obtain baseline results.

Stage2
As it is recognized, the fixed context length of LLM presents difficulties in handling extensive contexts, particularly in the case of biomedical data. To address this challenge, we employed a mechanism known as Selective Context, that makes the context simpler and more efficient. So, we employed this mechanism to compress the long context data to the LLM's max token length window and trained the LLM.

II. EXPLANATION OF THE SOLUTION

A. Selective Context

Selective Context utilizes self-information to selectively filter out less informative content [4]. This approach aims to improve the efficiency of dealing with the constraints of the fixed context length. Utilizing a base language model, Selective Context gauges the information content within tokens or token groups in each context. In our case, we used GPT2 to obtain the self-information values of tokens. It then keeps the most informative tokens and removes the less important parts.

B. Self-Information

Self-Information is a concept in information theory quantifying the amount of information conveyed by an event. In the realm of language modeling, this event pertains to an individual generation step, precisely a token [4].

Given a context {x_0,x_1..,x_n}, Self-information of a token x_i can be calculated as the negative log likelihood of the token [4],

$$I(x_i) = -\log_2 P(x_i|x_0, x_1, ..., x_{i-1})$$

Where x_i *is* a token, I(x_i) is the self-information and P(x_i | x_0 ,..x_{i-1}) is the output probability of the token x_i . In our case, we employed GPT2 to calculate the output probabilities of each token that in turn helped in calculating the self-information of each token.

Tokens with higher probabilities have lower self-information. We want to remove lower perplexity tokens or tokens with higher output probabilities as these tokens have less informative context (Commonly observed tokens).

C. Retention of Tokens

Upon computing the self-information for each token, the selective context method evaluates their informativeness. Subsequently, it dynamically selects the most informative content using a percentile-based filtering strategy. Initially, the tokens are ranked in descending order based on their self-information values. Following this, it identifies the p-th percentile of self-information values among all tokens [4].

Subsequently, the method selectively retains tokens with self-information values greater than or equal to the p-th percentile, creating a filtered context from the original context.

D. Selective Context API

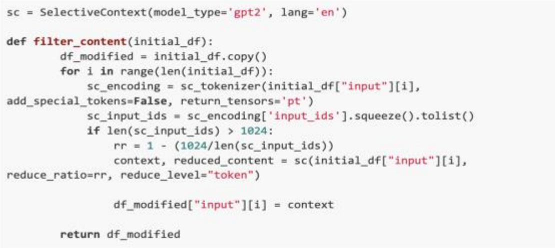
We utilized the Selective Context API to compress the input instances in our data, enabling two key operations: 1. Computing the self-information values and 2. Filtering out less informative context. Iteratively converted all the input data instances into compressed instances. Used different 'p'(percentile) values for different data instances dynamically.

We defined 'p' as follows,

```
p = (1 - (max_token_length_of_model/instance_token_length))*100
  = (1 - (1024/instance_token_length))*100.
```

Where, 'p' is the percentile value to filter out all the tokens corresponding to self-information values less than this p-th percentile self-information value. 'max_token_length_of_model' is the max token length allowed by the LLMs chosen (In our case, we have chosen In-BoXBART. It has the max allowable limit of 1024 tokens). 'insatnce_token_length' is the token length of an input instance in the dataset.

The function we defined to iteratively convert long context data instance into a compressed instance is as follows,



```
sc = SelectiveContext(model_type='gpt2', lang='en')

def filter_content(initial_df):
        df_modified = initial_df.copy()
        for i in range(len(initial_df)):
            sc_encoding = sc_tokenizer(initial_df["input"][i],
add_special_tokens=False, return_tensors='pt')
            sc_input_ids = sc_encoding['input_ids'].squeeze().tolist()
            if len(sc_input_ids) > 1024:
                rr = 1 - (1024/len(sc_input_ids))
                context, reduced_content = sc(initial_df["input"][i],
reduce_ratio=rr, reduce_level="token")

                df_modified["input"][i] = context

        return df_modified
```

Figure1: Function to compress the long context data

In this filter_content function, the parameter that is responsible for dynamically changing the 'p' (percentile) is 'rr' (reduction ratio). Here, rr = p/100.

The 'filter_content' function takes an initial data frame with long context data instances as input, and it iteratively compresses each data instance to LLM's max allowable token length based on the dynamic 'rr' value and replaces the original long context with compressed context. This function returns the modified data frame which will have the compressed context instances and this modified data frame is now used to train the LLMs.

III. RESULTS

In both the approaches, we set the training epochs to three.

A. Baseline Results

The results obtained during the stage1 of the project using the baseline approach where we have decided to truncate the input data to align with the maximum input token length of each LLM(In-BoXBART).

Table1.1

Dataset	In-BoXBART(Accuracy)
Smoking Challenge Data 2006	56.7308
Obesity Challenge Data 2008	71.86
Heart Disease 2014	48.3307

B. Results with Selective Context mechanism

The results obtained during the stage2 of the project upon training the In-BoXBART with the efficient and compressed contexts obtained from selective context mechanism.

Table1.2

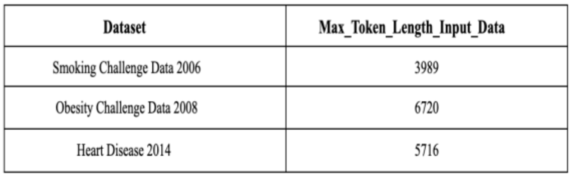
Dataset	In-BoXBART(Accuracy)
Smoking Challenge Data 2006	60.5769 ⬆
Obesity Challenge Data 2008	81.3096 ⬆
Heart Disease 2014	50.1987 ⬆

From the Table1.1 and Table1.2, we can see that there is increase in accuracies with In-BoXBART on all the three datasets compared to the baseline results. There is 3.85% increase in accuracy on Smoking Challenge Data 2006 [2], 9.45% increase in accuracy on Obesity Challenge Data 2008 [3] and 1.87% increase in accuracy on Heart Disease 2014 Data.

IV. CONTRIBUTION

In stage1, I visualized the data and analyzed the max token length possible from each of the datasets: Smoking Challenge Data 2006, Obesity Challenge Data 2008 and Heart Disease 2014 Data. I used In-BoXBART tokenizer to tokenize the data instances and found out the max token length possible from each of the datasets.

Table2.1



Dataset	Max_Token_Length_Input_Data
Smoking Challenge Data 2006	3989
Obesity Challenge Data 2008	6720
Heart Disease 2014	5716

Following the data visualization, I prepared the csv files corresponding to the train, dev, test datasets to feed the In-BoXBART. Also, I contributed in modifying the encoder-decoder script to make it suitable to our task which is a classification task. Following that, I trained the In-BoXBART model with Smoking Challenge Data 2006 for 3 epochs on the SOL supercomputer.

In stage2, I contributed to the literature survey to find out a method that can make use of all the data in an instance without discarding because of the bottleneck of max token allowable limit of LLM. After we decided to incorporate the Selective Context mechanism, I contributed to modifying the previous encoder decoder script to incorporate selective context mechanism after installing the requirements related to selective context in the environment. I designed the script structure in such a way that the initial data frames corresponding to train, dev, test datasets with long context data instances are passed to 'filter_content' function which returns the modified data frames containing the compressed data instances that later will be fed to the LLM.

I trained the In-BoXBART [1] model with 'Smoking Challenge Data 2006' [2] and 'Obesity Challenge Data 2008' [3] after incorporating selective context mechanism on SOL. The results are listed in Table 1.2.

V. LESSONS LEARNED

After actively engaging in the project, I have expanded my knowledge on working with LLMs. Following are the important learnings of mine,

- Learnt how the training of an encoder-decoder (It is In-BoXBART in our case) model works.
- Understood multiple ways to compress the long context data in an efficient way through the literature survey.
- Learnt how to utilize ASU's supercomputer 'SOL' to make use of GPUs for faster training of the model.
- Learnt the process of research and approaching a research problem through this project.

Team Members

- **Nagacharan Vemula(nvemula4@asu.edu)**
- Maaz Khazi(mkhazi@asu.edu)
- Ashish Ambadas Wale(awale1@asu.edu)
- Koushik Sai Achyuth Ayila(kayila@asu.edu)
- Hari Sai Charan Challa(hchalla2@asu.edu)

REFERENCES

[1] Mihir Parmar, Swaroop Mishra, Mirali Purohit, Man Luo, Murad Mohammad, and Chitta Baral. InBoXBART: Get instructions into biomedical multi-task learning. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 112–128, Seattle, United States, July 2022. Association for Computational Linguistics.

[2] Özlem Uzuner, Ira Goldstein, Yuan Luo, Isaac Kohane, Identifying Patient Smoking Status from Medical Discharge Records, *Journal of the American Medical Informatics Association*, Volume 15, Issue 1, January 2008, Pages 14–24, https://doi.org/10.1197/jamia.M2408

[3] Özlem Uzuner, Recognizing Obesity and Comorbidities in Sparse Data, *Journal of the American Medical Informatics Association*, Volume 16, Issue 4, July 2009, Pages 561–570, https://doi.org/10.1197/jamia.M3115

[4] Yucheng Li. 2023. Unlocking context constraints of llms: Enhancing context efficiency of llms with selfinformation-based content filtering. ArXiv preprint, abs/2304.12102.