GOOGLE SUMMER OF CODE

ABOUT-ME:

NAME: Sanjay Srikanth

EMAIL: sanjay.sriksun@gmail.com

College: New Horizon College of Engineering

YEAR: 2

LOCATION: India (Time zone GMT+5:30)

SKILLS: Python, C, basics of OOPs and java

PROBLEM: **Add an AI chatbot to the Chat Activity**

PROFILE: https://www.linkedin.com/in/sanjay-sriksun/

WHY THIS PROJECT:

As a beginner to the AI and ML course stream, the '**Add an AI chatbot to the Chat Activity**' problem statement has piqued my interest. Sugar Labs presents an excellent platform for facilitating children's learning across different skills through various activities. This has caught my interest, giving me an opportunity to contribute to this project.

The chat activity within Sugar Labs holds significant potential as a fundamental communication tool for children. It enhances a child's communication skills and their vocabulary.

Reasons why I would like to contribute to this project:

1. I have always found interest in AI and I believe this is a great opportunity for me to do something in this field.

2. The chatbot to the chat activity is a wonderful idea for children to increase their communication skills and vocabulary.

3. This will be a learning experience for me in the field of AI and I hope to understand better on how language models truly work.

Besides this, I would like to build this chatbot as a complete useful asset for the children apart from my learning experience.

HOW WILL SUGAR LABS BE AFFECTED?

1. It will enable general engagement between children and Sugar Labs.
2. It will also enhance children's communication capabilities.
3. Sugar Labs can now be a far better tool for a child and boost their skills.
4. Young users can learn basic words, how to speak in formal and informal ways.

APPROACH TO PROBLEM:

The chat activity is going to be based on fine tuning an already existing pre-trained language model. There exists a couple of free and open LLMs that can be fine-tuned. **Pytorch** offers a pre-trained language model to work with. I intend to use Pytorch to fine tune the model with different datasets and build a new model architecture for the model that aims for the younger audience.

Project Gutenberg can be used as a dataset since it has a variety of e-books, it's files are free to use and is a large text corpus. Moreover, the content is in plain text format which makes it a better source for a machine to learn. I intend to establish a good engagement of conversation and since e-books have various content, I believe it will be easier for the model to train on this dataset. Hence, I will use Project Gutenberg for machine learning and text generation.

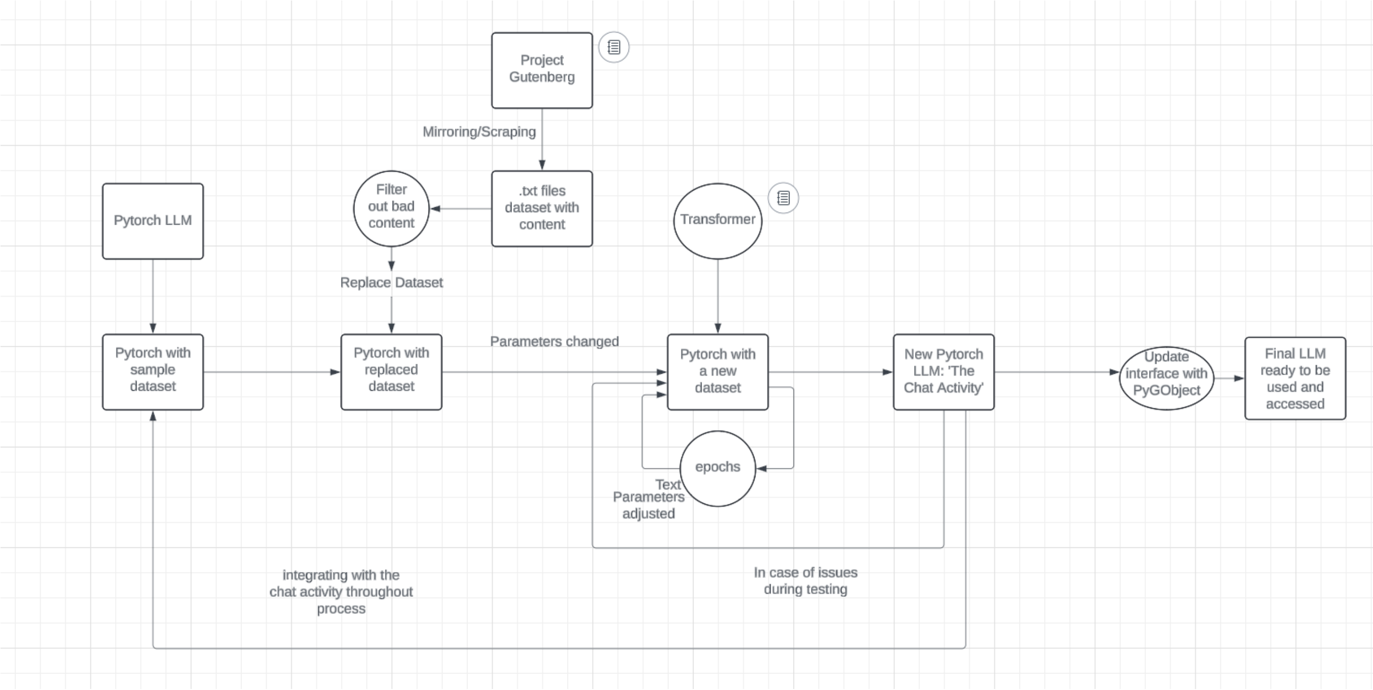
Any inappropriate content will be filtered out and the model will be trained. The model will be trained on unsupervised and reinforced learning. This is to ensure better response inputs and outputs based on feedback. Once trained, the model gets integrated to the chat activity.

I intend to use **TEXT files** to store all the data for the dataset. I also have knowledge on **python** which I would most likely use for the project.

I will also be building a custom transformer using Pytorch for the project that will train on the text files.

I hope my skills are enough to contribute and I'd love working on this project.

Given below is a block-chart diagram of how the LLM is being created and trained:


Project Gutenberg
Mirroring/Scraping
.txt files dataset with content
Filter out bad content
Replace Dataset
Pytorch LLM
Transformer
Pytorch with sample dataset
Pytorch with replaced dataset
Parameters changed
Pytorch with a new dataset
New Pytorch LLM: 'The Chat Activity'
Update interface with PyGObject
Final LLM ready to be used and accessed
epochs
Text Parameters adjusted
integrating with the chat activity throughout process
In case of issues during testing

LINK: https://lucid.app/lucidchart/36fb63c9-8f04-49c2-a554-60b4f98d169e/edit?viewport_loc=-56%2C-97%2C2219%2C1076%2C0_0&invitationId=inv_caabbd63-d599-4ea0-bfb8-1ac8b6e7d81a

(Lucid website for creating block diagrams and flow charts)

TIMELINE:

I plan to adhere closely to this timeline for the project, ensuring thorough follow-through to complete the project and fulfill my contribution.

The 175-hour project will be divided in weeks as follows:

Additional Note: I will be working on building small custom LLMs throughout April to familiarize myself with the working of LLM and will do necessary installations for the same. Getting to know the process would boost me far better in doing the project.

May (4 weeks):

May 1-10 -> Community bonding. Here I will also work on fixing a couple bugs on some Sugar Labs activities to gain some knowledge on how the community works.

May 10-20 -> Going through the chat activity codebase and the Pytorch source Code. I will also build a very small dataset as a sample.

May 20-31 -> Building a custom transformer to train on a very small dataset and integrate with the chat activity. This makes it clear on how my final model should be like.

June (4 weeks):

June 1-14 -> Creating actual dataset in .txt files mirrored from Project Gutenberg and replace it with the small dataset created before.

June 14-20 -> Using the same transformer earlier with changed parameters. This transformer will now train on the new dataset created. May take couple days.

June 20-25 -> The model is tested for performances, if there's any issues, the dataset is checked again.

June 25-30 -> The dataset is fine-tuned again to remove any inappropriate content.

Throughout June, I will iterate on the chat activity integration code to get it in shape.

July (4 weeks):

July 1-5 -> The testing phase begins and the model is checked for its working status.

July 5-9 -> Fixing bugs or any issues that occur.

July 9-10 -> Submission for mid-term evaluation.

July 10-31 -> I will work on a better interface for the model that aligns with the interface of the chat activity. If any changes are required for the chat activity, will handle them through additional PRs. The chat activity is in GTK +3 version. I will use PyGObject to work on the interface.

August (4 weeks):

August 1-10 -> Fixing bugs and issues, testing out the final model and making any changes if necessary. Will prepare documentation around the feature.

August 10-19 -> Testing with actual younger audience and looking for feedback and make changes if needed.

August 20-22 -> Final review and submission.

During the course of the project, at the end of every week my progress will be shown to the mentor and the timeline will be followed strictly.

NOTE: I depend on Sugar Labs for hardware for LLM training.