**a PART I.**

**a. General Information**

Project Title: Exploring Question and Answering Techniques using BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-Trained)

Submitted by (insert names of team members): George Kaceli and Nisarg Patel

Submitted to (supervisor name): Jianguo Lu

Date submitted: March 10th, 2023

**b. Project Overview:** *Describe the project & its purpose**(300- 400 words)*

The goal of this study is to investigate and contrast the performance of two innovative language models, the BERT and GPT transformers, using question-answering (QA) techniques on the SQUAD 2.0 dataset. The project will include an introduction to the transformers, training them on the dataset, and optimizing them.  
   
Two of the most well-known language models are BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), which were created by Google and OpenAI, respectively. BERT is an encoder-only transformer that creates contextualized embeddings for each input token using bidirectional self-attention. GPT, however, is a decoder-only transformer that creates text one token at a time using autoregressive language modeling, on multiple NLP (Natural Language Processing) tasks, including QA, both models have produced state-of-the-art results.

Furthermore, several experiments have been conducted concerning the BERT language model to improve our approach, including analyzing the tokens in the larger BERT model and running BERT on a sample paragraph to get a better understanding of how the model works.

About 100,000 queries were submitted by crowdworkers on a collection of Wikipedia articles as part of the SQUAD 2.0 dataset, a benchmark dataset for quality assurance. Answering these questions requires choosing a section of text from the appropriate article. The dataset is divided into training, development, and test sets, with the training set having over 130,000 cases and the development set having 12,000 examples.  
   
We will train and fine-tune both models using the training set before comparing their performances on the dataset. By minimizing a loss function that penalizes wrong responses, the models are trained on quality assurance during fine-tuning. The models' hyperparameters will be tuned using the development set, and the top models will be chosen based on their F1 score and exact match accuracy.

When the models have been trained and adjusted, we will assess how well they performed on the test set and contrast the outcomes. To determine the kinds of questions on which each model excels and those on which it suffers, we will also conduct an error analysis. Finally, we will talk about the learnings from this project and how these methods may be used in practical situations.  
   
In conclusion, the dataset will be used in this study to examine and contrast the methods employed in BERT and GPT transformers. The project will involve evaluating the models' performance, training, and fine-tuning them, and conducting an error analysis.

**c. Deliverables:** *Describe all products to be produced* *(200- 300 words)*

This project really can be summarized in two or three phases, in the first phase we will implement a BERT language model and fine-tune it for our purposes, then do the same with a GPT language model and then compare the two.

In the first phase of the project, we will take a pre-existing BERT language model and fine tune it to our needs. In this case we will look at the minimum to fine tune BERT for question answering and then look to see if we can build any more applications on top of it. But at minimum we will have to deliver a fine-tuned BERT language model for question answering.

Furthermore, we will do the same thing with a free open-source GPT language model after we have completed the implementation of BERT, we will be expected to deliver a GPT language model that is fine-tuned for question answering and another application we choose to build on top of it. It must be noted that any applications we build for BERT must also be built for our fine-tuned GPT language model as we will be comparing the two in the next phase. We will also try to incorporate the LLaMA model, another large language model created by the team at Facebook which competes with BERT and GPT. Our aim is to compare and analyze the bottleneck each model presents.

Lastly, we are expected to evaluate each model's performance, compare them with one another, and highlight our findings in a presentation that offers an insight into both models and how they work. Then we will compare their performance based on a set of variables used for the evaluation. These variables might be things such as error rate, success rate and others to figure out what each language model has to offer.

**d. Requirements:** *Describe the required resources e.g., hw/sw, technical knowledge and skills etc*. *(100- 200 words)*

The required resources of this project are straightforward from a hardware perspective, nothing we are implementing is so heavy in quantity of data or complexity that the implementation and testing cannot be done on our personal machines.

Furthermore, in terms of software, resources such as PyTorch, TensorFlow will be used for the machine learning aspect of this project, as well as having Python installed, a version of Python 3 seems to be the most optimal as Python 2 will not suffice. Furthermore, other resources such as Jupyter Notebook can be quite useful as well for our purposes.

Lastly, on a technical level, a knowledge of Python is necessary as much of the applications that we are building and our language models are implemented using Python libraries and the Python programming language, so for our purposes some prior knowledge in Python is required.

**PART II.**

**1. Risk analysis** *(There is no fixed format, as it will vary depending on the nature of the project. For some general guidelines please see the file under the* ***ResourcesàRisks*** *link on BB)*  
  
The following are the risk analysis for the described project:

* Technical risk: There is a chance that the project will need a lot of technical knowledge in software development, programming, and Natural Language Processing (NLP) to build a system that can correctly respond to inquiries from a given dataset. The experiments might also need sophisticated gear or cloud computing resources.

* Data risk: There is a chance that the dataset used to train and test the models will not be sufficiently diverse, which could result in skewed findings. Also, there is a chance that the models may perform poorly due to insufficient or noisy data.

* Time risk: There is a chance that the project will take longer than anticipated owing to unforeseen difficulties, such as faulty technology or software, inconsistent data, or a lack of knowledge in a specific field.

* Performance risk: Due to the complexity of NLP tasks, there is a chance that the built models will not perform as predicted or that the comparison will not produce definite results.

The following techniques can be used to reduce these risks:

* Technical risk: To reduce the amount of code necessary, investigate and utilize existing libraries and frameworks. To meet hardware requirements, think about leveraging cloud computing resources.

* Data risk: Get a variety of pertinent data for the models' training and evaluation. Make sure the data is accurate and adequate for the purpose. To expand the dataset's size and diversity, use data augmentation approaches.

* Time risk: Provide enough buffer time in the project plan to account for unforeseen difficulties. To make the best use of team members' time, divide the project into smaller jobs and distribute them among them.
* Performance risk: Do exhaustive experiments and make sure the findings are statistically significant. Verify the effectiveness of the models using current benchmarks and evaluation measures.

**PART III.**

**1. Work Breakdown Structure (WBS)** *(There is no fixed format, as it will vary depending on the nature of the project. For some general guidelines please see the file under the* ***Resourcesà* *Work Breakdown Structure (WBS)*** *link on BB)*

**For Part A):**

**Goal:** Fine tune a pre trained BERT and GPT model for question and answering

**Activities:** Need to do prepare dataset, load pre trained model, fine-tune, evaluate and test models' ability to make predictions.

**March:**

**Activity 1: Prepare Dataset**

Tasks:

* Find the dataset to use to train and fine tune the model
* Use the SQuAD v2.0 dataset as it is the benchmark for question and answering
* Download the dataset
* Divide the dataset into training and validation set

**Activity 2: Load Model (BERT)**

Tasks:

* Load the Pre-Trained Model by installing the Hugging Face Transformers model in Python to do so
* Make sure to initially utilize the smaller BERT model.
* Also load the corresponding tokenizer as well.

**Activity 3: Fine Tune the Model (BERT)**

Tasks:

* To fine tune our model (BERT) we can use the train method in the Hugging Face library and try to optimize our model
* We will have to adjust parameter and play around a bit to improve the performance
* May involve fine tuning for more epochs
* Increasing or decreasing training data size
* Adjusting learning rate, optimizer

**April:**

**Activity 3:**

Tasks:

* May continue into April with fine tuning

**Activity 4: Evaluate the model (BERT)**

Tasks:

* Using the validation set evaluate the performance of the model
* We can use the eval method in the Hugging Face library
* To do this we must calculate the following:
* Accuracy, F1 score and exact match score
* If our model performs up to par continue,
* If not go back to Activity 3

**Activity 5: Test Models ability to make predictions.**

Tasks:

* Prepare the input example
* Feed it to the model and get predicted output
* Interpret the output and see if model predicted correctly
* If not repeat activities 3 and 4

**For Part B):**

**September:**

**Activity 7: Load Model (GPT)**

Tasks:

* Load the Pre-Trained Model by installing the Transformers model in Python to do so
* Make sure to initially utilize the correct GPT model.
* Try to utilize the LLaMA model, which is open source to academia, and got leaked recently
* Also load the corresponding tokenizer as well.

**Activity 8: Fine Tune the Model (GPT)**

Tasks:

* Data preprocessing
* Model Fine-tuning
* We will have to adjust the parameters and play around a bit to improve the performance.

**October:**

**Activity 9: Evaluate the model (GPT)**

Tasks:

* We will evaluate the performance of the model on various metrics.
* Including accuracy precision, recall, and F1 score.

**Activity 10: Test Models (GPT) ability to make predictions.**

Tasks:

* Prepare the input example.
* Feed it to the model and get predicted output.
* Interpret the output and see if the model predicted correctly.

**November:**   
**Activity 11: Comparing all the findings from both the models.**

Tasks:

* Comparing results of both BERT and GPT based on accuracy and various other factors.

**Activity 12: Analyze the different behaviors of the model from the findings.**

Tasks:

* Analyzing how both models interpret the same question and comparing the generated answers.
* Analyzing the tone of the answers produced

**December:**

**Activity 13: Work on the presentation**

Tasks:

* Should be in .pptx format
* Will explain each of the models and explain what each do
* Will show our findings and the performance of both models
* Display our contribution and highlight further work that could be done