AI Thesis Helper

by

Mirza Ahmad Shayer
22273008

A project submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Engg. in Computer Science and Engineering

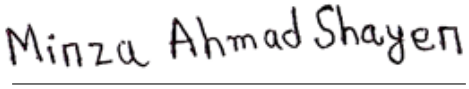
Department of Computer Science and Engineering
Brac University
November 2024

Declaration

It is hereby declared that

1. The project submitted is my own original work while completing degree at Brac University.
2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

Student's Full Name & Signature:



Mirza Ahmad Shayer
22273008

Approval

The project titled "AI Thesis Helper" submitted by

1. Mirza Ahmad Shayer (22273008)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Engg. in Computer Science and Engineering on 14th November, 2024.

Examining Committee:

Supervisor:
(Member)

Moin Mostakim
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md Sadek Ferdous, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In the modern world, Artificial Intelligence (AI) has become popular. Machine Learning (ML), is another popular term used on the internet, it is when AI is applied to a system which allows it to learn. Deep Learning (DL) is another popular term that became very common in recent times, it is the application of ML that utilizes algorithms and trains models. In 2020, Generative Pre-trained Transformer (GPT), came to the mainstream and had become highly successful. In just another year it became a game changing tool in the scientific landscape. Using GPT, various chat-bots are designed. Chat-GPT in particular is by far the most used and most useful model of GPT. The model is useful in generating replies to various questions with high accuracy and usefulness. Can be used to help users with almost practically any question. In a few years this model will develop to help people even further. As such, this project - AI Thesis Helper, is one such implementation of further helping students to aid their thesis writing. Many students struggle to write their thesis, as such this chat-bot can help students with various thesis tasks like idea generation. This project aims to assist students in their writing. Project challenges and objectives are elaborated. Prior works are looked into along with reasons for choosing GPT over other models. Requirements, pros, cons and feasibility - both economic and technical along with model diagrams are dealt with. Details on the dataset are discussed. Fine-tuning of the model is elaborated with all details like imports, hyperparameters and training loop etc. Front-end, integration and back-end are discussed along with UI/UX components. Implementation shows the project workings and how well it performs. To evaluate the performance, the following metrics are utilized - model loss during training, precision, recall, F1 score, perplexity and spelling, grammar with overall writing quality. Along with a comparison of generated text, made with two other AI chatbots - ChatGPT and Perplexity AI for the same text input. Future updates are also described.

Keywords: Deep Learning (DL), Machine Learning (ML), Generative Pre-trained Transformer (GPT), Artificial Intelligence (AI), chat-bot, Chat-GPT, GPT-2, PyTorch, Gradio.

Acknowledgement

Firstly, all praise to Allah for whom my project has been completed without any major interruption. Secondly, to my Supervisor Md. Moin Mostakim sir for his kind support and advice in my work. He helped me whenever I needed help. And finally, to my parents, without their thorough support it may not have been possible. With their kind support and prayers I am now on the verge of my graduation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the few recent years, technology has turned its focus to task automation. A system can learn to do the task on it's own without the aid or input of any human being. From 2020 and onwards, various technology companies have started to build machine algorithms that can perform complex tasks in a matter of a few hours or even minutes. As such, Artificial Intelligence (AI) took center stage in the technological development process. From AI, came new systems that could learn tasks, words, patterns or detect similarities. This came to become known as Machine Learning (ML). From ML onwards, very specific algorithms for very peculiar purposes were being built up. This is known as Deep Learning (DL). From 2023, "Transformers" became mainstream in deep learning. "Transformers" are types of neural network architectures that changes an input sequence to an output sequence. From the model various chatbots were designed. Generative Pre-trained Transformer (GPT) became the most popular. From GPT, Chat-GPT had been designed. It became a phenomenal game changing tool in the world. It is able to answer questions shortly with good accuracy. Building a chat-bot for a specific task can mitigate many problems with general chat-bots. According to Vaswani et al., an attention system is utilized to connect the encoder and decoder for the best models. The "Transformer" is used with only the attention system and dropping off the recurrence and convolutions completely. These models are superior and provide more accurate results. Requires less time to train and can perform various multiple tasks with great efficiency [20]. According to Pereira et al., writing helps in the development of various meticulous skills. General Artificial Intelligence (GAI) are now getting very common even in the academic writing sphere. This can have a lasting impact and all sides must come to consideration of potential issues [29]. During university, many students must submit a thesis to complete their undergrad or post-grad degrees. Students always seem to struggle with writing thesis. Confused about, formats, ideas, where to begin, what to look for and where to find resources. As such, this project is motivated to bring an end to these problems. The AI Thesis Helper will aid students with all the mentioned issues faced during any thesis. This specific chat-bot will be designed for thesis students to give the utmost correct and accurate answers for questions students may ask.

1.2 Problem Discussion

Often times, university students always struggle with writing the thesis. Most often, what format to use or follow, where to get open access resources, where to find codes or code templates, what formats to use for citations or references along with a myriad of other issues. Very often spelling and grammatical errors are also present in student thesis. According to Park, among teachers and English as a Foreign Language (EFL) learners computer assisted language learning has become very popular. English grammar checkers in particular have made large contributions to maximizing learners ability to learn while also reducing workloads on teachers [5]. Again according to Wong et al., in this era of globalization, technology has been a pivotal learning tool in case of language learning. In professional and academic success, clear and correct writing is necessary. Most often learning a foreign language often leads to various writing mistakes. Mobile spelling checkers have now become more common and are often utilized to decent success to solve these errors [12]. According to John and Woll, the study done assessed the performance between current popular spelling checkers - Grammerly and Virtual Writing Tutor. Most often spell checkers have only about 50% correct results. The study shows that both Grammerly and Virtual Tutor Point have slightly more accuracy than Microsoft Word [3]. As such, students facing such issues will slow down their ability to finish thesis on time and will also cause various stress. Thesis writing is also difficult as students must also be already well articulated with various writing, noting, structuring and factual skills. Not all students have such skill and thus it becomes a hassle.

1.3 Project Aims

With thesis students in mind, the aims of this project is to ensure a good chat-bot is made that can help students with any kind of thesis related question. Students can ask the chat-bot for tips, help, advice and can receive general instructions on how to move forward with given topics. Secondly, the answers generated by this chat-bot will be more accurate to the given topic inputted by students. Students can also ask for more information or more accurate answer formulation. Chat-bot will also deliver answers in simple manner. Chat-bot can give alternative answers if students ask for them. It will also explain parts of paper or summarize paper. Can give both generalized answers or more specialized answers if asked by students - requires more description and prompts. According to Rodriguiz and Mune, a chat-bot made by two graduate interns in the public university library saw more interaction as time went on. The people were intrigued to use the chat-bot after receiving very well given answers. Chat-bot used the resources within the public university library to generate all answers. Used peer-to-peer reviewed journals and papers, books and various magazines or articles [13].

1.4 Project Objectives

The objectives of this project is to build an AI Thesis Helper Chat-bot. The chat bot will be designed to give students with answers, formats and various other necessary information. Can explain or summarize papers, paragraphs and any given

inputs. Can paraphrase sentences. Chat-bot will have a simple intractable interface, somewhat similar to ChatGPT. Customizable interface for the AI Thesis Helper. Spelling check and grammar check of various sentences. According to Hládek, Staš and Pleva, there is now a serious research interest in automatic spell checking and grammar. There is still a lack of a proper understanding framework for the automation. Natural Language Processing (NLP) requires normalized forms of words to work effectively. If words are jumbled up, wrong or incorrect there is a chance of error in the answer generation. Interactive systems underline wrong words and suggests corrections [7]. Overall these are some of the objectives of this project.

The summary of the objectives for the AI Thesis Helper are given below.

1. Answers student questions.
2. Give explanations.
3. Summarize and paraphrase paragraphs.
4. Explain paragraphs and other information.
5. Spelling and grammar checking.
6. Simple intractable interface.
7. Customizble interface.

1.5 Project Challenges

This project came with a set of challenges that had to be overcome to some degree. Firstly, finding an open access data set to train the machine on with an adequate amount of resources was some what difficult to find - for this specific purpose. Luckily there was a public dataset available called "Arxiv-100" which was later modified for the task. It has all the necessary data that is required to train the machine. Secondly, which framework to use. After much thought and consideration - PyTorch frameworks were chosen for their versatility, availability and flexibility. The frameworks were also rather easy to work with and great amount of resources and documentation were available online. Thirdly, which Integrated Developmental Environment (IDE) to use for this specific project. After some brainstorming on the amount of resources required, available resources and difficulty of implementation - Google Colaboratory was selected for the machine training. It had all the features and datasets uploaded, and gives access to back-end GPUs. The project would be done in Python. For the UI/UX interface HTML and CSS would be utilized and would be done in VSCode IDE. Back-end would also be done on colab. Gradio interface framework would be utilized for the integration. Finally, one challenge that persisted was the number of epochs the machine could run for. Due to Colab's limitation on GPU usage and time, the number of epochs would be limited. This may affect the outcome in certain cases.

1.6 Organization of the Report

This report is organized into various chapters, each explaining the contents of the chapters. The chapters are organized and all details are explained very colloquially.

- Chapter 1 - Discusses the motivation, problem discussion, project aims, project objectives and project challenges in detail. Along with report organization.
- Chapter 2 - Elaborates the literature review needed for this project along with related or previous works. Also sheds light on why GPT model is chosen over other various architectures also due to being a decoder model.
- Chapter 3 - Explains requirement analysis, computational requirements, pros, cons and feasibility report both economic and technical along with model diagrams of the given project in detail.
- Chapter 4 - Explains the project workings, the dataset used, training the machine, front and back end development, integration along with UI/UX components all explained in detail.
- Chapter 5 - Discusses project implementation and how well the machine performs. How well the components of the machine function and model performance through various metrics are elaborated here.
- Chapter 6 - Talks about the conclusion and future updates that will be available to the current project. Also updates featuring more functions will be discussed as well.

Chapter 2

Literature Review and Related Works

2.1 Literature Review

In these couple of years, computers and machines have become very developed. At the start of 2020, artificial intelligence has become very mainstream, and with that came machine learning and then later deep learning was developed. In 2022 ChatGPT, became extremely popular in a very short span of time. GPT is a transformer. Transformers are more powerful than neural networks. Although transformers are now used mostly for Natural Language Processing (NLP) tasks. With the recent surge of transformers many other types of GPTs are being made. According to Yang, Zhu, Gai and Wan, in NLP the discovering of semantic patterns is a core property. It helps in sentence formation and vocabulary usage. It serves as an important function for text classification. This area of research is now thriving. Now with pre-trained models like BERT, RoBERTa and GPT-3. Word2vec model is a prime example of the best word embedding model [33]. As per Noroozi et al., to address the limited sampling of cross-modal relationships between text domains and speech in samples is an important tactic to upgrade performance.

Large Language Models (LLMs) are utilized to make textual components and text-to-speech models to generate parts of speech. These methods provide a more practical and better means of enlarging the training datasets [28]. As per Gaikwad, Iyer, Talluri and Salve, various mental issues are growing in today's fast paced technological driven world. Utilizing text generation models like DialoGPT and T5 Transformers. The given model works using prompt engineering. This model can imitate naturalistic human speech and answer with humor to mimic human conversation. This helps in building rapport with the user and can also reduce stress in the user [24]. According to Tamrakar and Wani, for chatbots speech and textual language is the most common method of communication. Chatbots take the inputs of the language of the user and then generates smart, precise and humanistic replies for the given inputs. Chatbots have the added advantage of scalability to the number of users it interacts with in any communication system. Many companies now utilize chatbots in messaging for their users [9]. As per Adike, the critical prospect of a chatbots effectiveness is how well the conversation between the program and the user is held. Coherent talking flow is necessary for them to be very effective [14].

2.2 Related Works

Chatbots have now become the new normal for most online conversations. With the help of transformers data processing for NLP is now more easier and much faster. According to Turner, transformer is a component of a neural network that can learn important representations of data-point sequences. This has caused advances in the field of spatio-temporal modelling, computer vision and natural language processing. A more mathematical representation of this component is shown here. Transformers take in data and returns a representation sequence in another matrix of size DxN. The transformer block is given below by a simple equation.

$$X^{(m)} = transformer - block(X^{(m-1)}) \tag{2.1}$$

The block has two stages - one operates across the sequence and another across the features. First stage refines each feature separately, utilizing word patterns and positions. Second stage refines features representing each token, acts vertically across a column. This structure can also be part of a more complex system due to the encoding-decoding model [32]. The transformer architecture is also complex when compared to other neural network models. Below shows the images of a transformer model architecture and their special components.

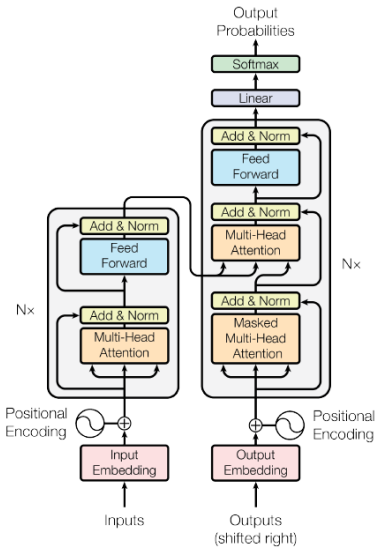


Figure 2.1: Transformer Model Architecture

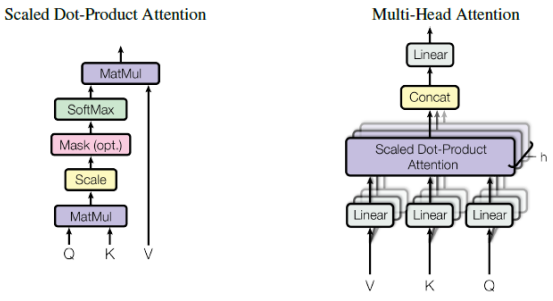


Figure 2.2: Scaled Dot-product attention (Left) and Multi-head attention (Right)

As per Figure 2.1, it shows the encoder and decoder stacks where both have $N = 6$. All the layers and details are shown in the diagram. Figure 2.2 shows the two attention mechanisms. In scaled dot-product, multiple queries are packed into a matrix Q. Keys and Values also packed into matrices K and V respectively. Multi-head attention allows to join different information from different representation sub-spaces at different points. These are the basic workings of a transformer model [20].

The use of chatbots, now in particular Chat-GPT has become rather very popular. In various fields, in some cases ChatGPT is utilized to solve various small problems, generate messages and help with basic tasks to reduce load and save time. In education, ChatGPT is now being utilized by both students and teachers to speed up the learning process. According to Nikitina and Ishchenko, ChatGPT has both positive and negative effects in education. When utilizing this for education, the practical significance and affects must be first understood before application is given the green light. Modern education requires innovation to stay relevant and interesting. ChatGPT is good at sentence generation but sometimes misses contexts. Various people have various attitudes towards ChatGPT usage in education [27]. After 2019, due to the coronavirus, for about 2 years most of the world went for an online only communication in many corporate and educational settings. According to Abdelmoiz, Mostafa and Soliman, the pandemic radically shifted the world to a more digital scene and thus created a high demand for online communication. This facilitated the advancing of natural language processing, which in turn also prompted for more chatbot creation to meet the demands. Many chatbots were made to handle commerce, e-learning, e-business, marketing and various other chat related tasks [21]. Chat-bots are also used for social engineering as well. As per Collier, human security is also the most vulnerable weak point. In some instances, chat-bots can be utilized to give users commands or orders that may violate laws or principles. Other times can be used to hack other systems or find vulnerable exploits done by malicious users. Information overload is another issue and many groups can weaponize chat-bots for their own purposes. Overall, both companies and users must be very aware of policies regarding such issues and use chat-bots very carefully. Also chat-bot can be designed with safeguards that stop people from making it ask or answer questions related to harming others [22].

Overall, chat-bots have advanced artificial intelligence very quickly. ChatGPT is now having effects in various different fields. As per Kalla, Smith, Kuraku and Samaah, ChatGPT has now been utilized in various different fields for various different tasks. ChatGPT can generate naturalistic human speech which is an advantage but, the disadvantages are that the answers may be biased. It is also very cost effective [18]. In education ChatGPT is very often utilized. According to Çalışkan, ChatGPT is now very often utilized to make course materials for students to learn. It saves a lot of time and often the course materials generated are also very helpful. Sometimes lectures and presentations often have few grammatical mistakes and often times does not understand context. Overtime there has been some improvements though more is necessary [16]. Chatbots have now made online conversations easier and more efficient. Also helped in customer services and speaking with clients worldwide.

2.3 Why GPT Over Other Models

For chatbots, there are many models available. Models to make the chatbot are dependent on what task the chatbot needs to do. The model is chosen based on what dataset it will train on, the results it will produce and what task the model will perform. Below shows a diagram of various encoder-decoder models.

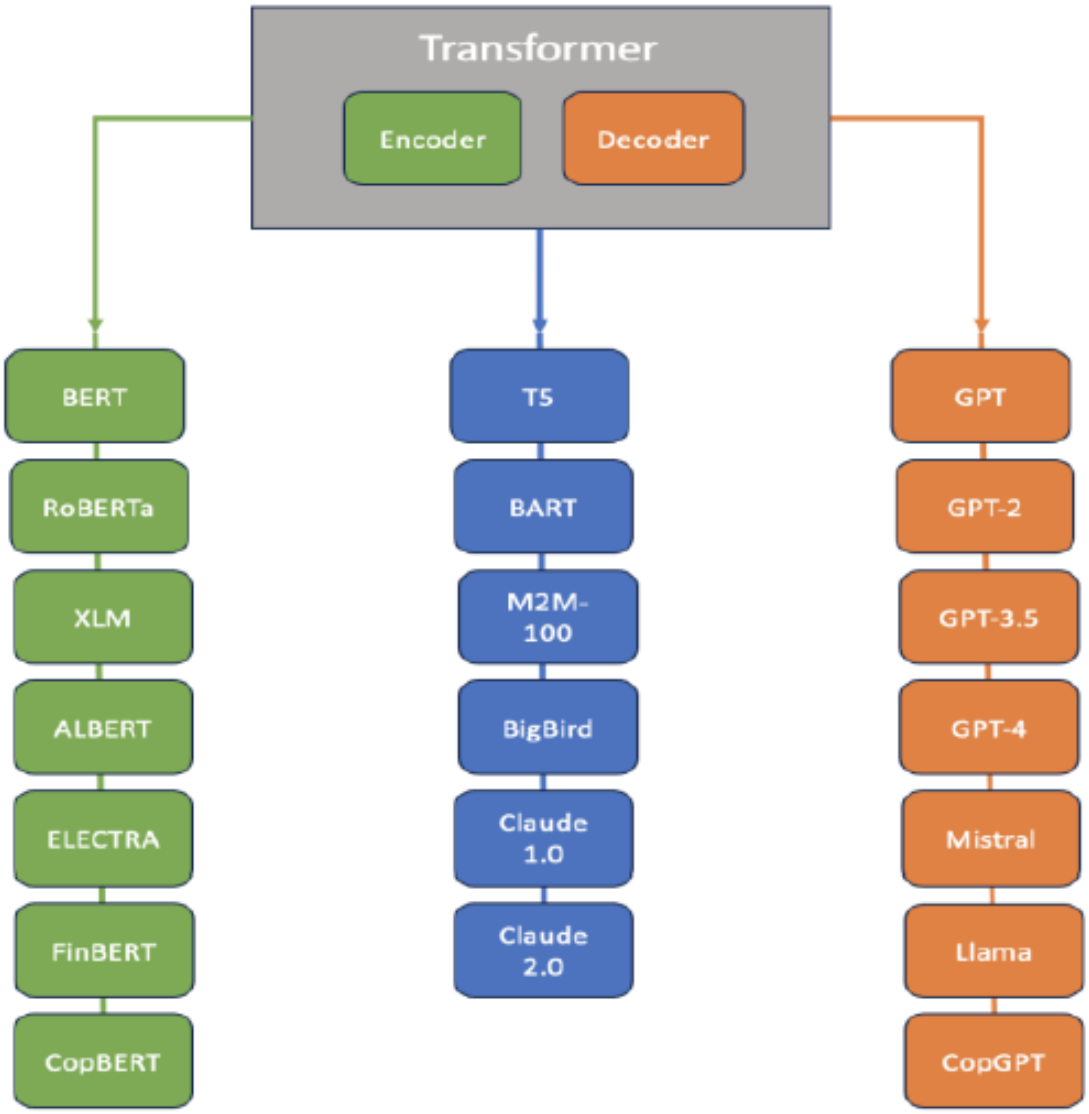


Figure 2.3: Various Encoder-Decoder Models

As per Figure 2.3, various models are made for various purposes. According to Sharkey and Treleaven, BERT and GPT are most often the best models for most NLP tasks. CopBERT and CopGPT are the best variants, though CopGPT is slightly more better than CopBERT. Larger LLM models outperform BERT models. GPT models are decoder types and in text generation these ones are preferred. Decoder models can handle text generation better due to its embeddings, layers and activations.

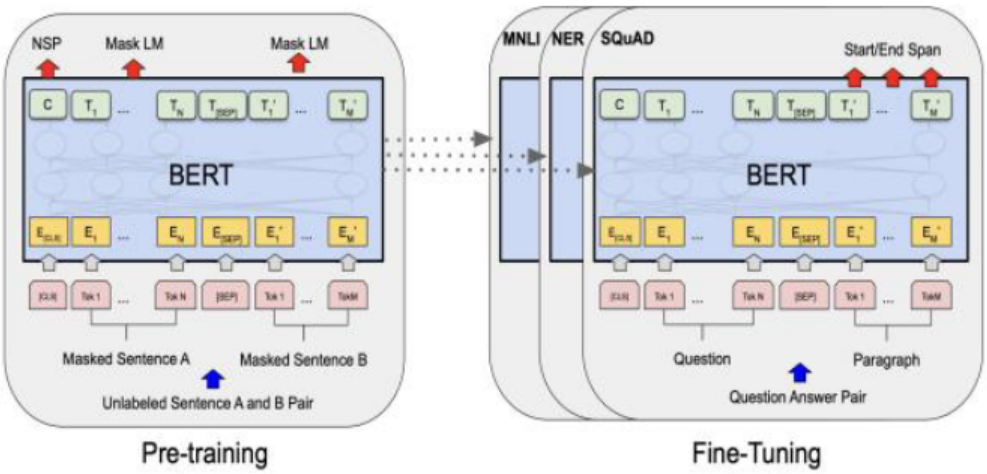


Figure 2.4: BERT Model Architecture

As per Figure 2.4, BERT is designed to pre-train deep representations by looking at both the left and right context in every layer, allowing it to learn from unlabeled text in a bidirectional manner. The pre-trained model can be fine-tuned with just an extra output layer to create the best models for many tasks. The results for all the models are shown below.

Table 2.1: Model Results

Model Name	F1-Score (%)
Random Baseline	0.23
FinBERT Sentiment	0.37
CopBERT Sentiment	0.41
CopGPT (GPT3.5 turbo finetuned) Sentiment	0.56
GPT4 Sentiment	0.49
GPT2 Sentiment	0.38

As per Table 2.1, CopGPT has highest F1-Score compared to the other models [30]. According to Zayyanu, BERT comprehends milieu of words in a sentence, GPT produces generative contextual texts and T5 treats all NLP tasks as text-to-text. All three have shown to have superior performance is many NLP based tasks. All three can be fine-tuned easily depending on the task [26]. For any chat-bot, generating naturalistic poems is the most difficult NLP task. If the chat-bot can do this then it can handle any generation task. According to Hämäläinen, Alnajjar and Poibeau, poem generation is a difficult task for most NLP models. Various different approaches have been applied to models to mitigate this limitation. Poems in older languages offer a bigger challenge [11]. According to Koubaa, GPT-4 has more parameters and can work with both texts and images, while GPT-3.5 can only work with text. GPT-4 is a higher large scale model compared to GPT-3.5. In some specific tasks GPT-3.5 is seen more superior than GPT-4 but not all tasks. Both models are still updating [17]. According to Liu et al., both ChatGPT and GPT-4 do well in logical reasoning, even with newer data sets. Though both struggle when it comes to NLP inference tasks that require logical reasoning [19]. For all these reasons, limitations and advantages GPT was chosen.

Chapter 3

Project Description

3.1 Requirement Analysis

The project is rather a simple application, which does not require users to have much knowledge of any computer expertise or machine learning knowledge. Any user can use this application. The application has peculiar requirements that need to be met. The application is rather easy to use and very beginner friendly for anyone. The requirements and details of the application are explained below.

3.1.1 User Requirements

There are some user requirements that need to be met for the user to utilize this application. The requirements are explained below in detail.

1. Users need to have internet access to use the application.
2. Users can go to the application using a URL to utilize the application.
3. Users then can start a new conversation and start writing and chatting. The application will generate answers.
4. Users can start a new conversation.
5. When users are done using the application, users can choose to close the application.

These are some basic user requirements for the application. If users follow the requirements, the application should run without any issues. Internet is necessary for the application.

3.1.2 System Requirements

The application requires a few system requirements to be met. For the application to work smoothly and without any hassle, these basic system requirements need to be met. The application is rather simple and does not require powerful systems to run. This application can run on any basic system. The necessary system requirements are given below in detail.

- Users should have a basic working laptop or desktop.

- Users need to use a browser to access the application.
- Users need an internet connection to use the application.
- Optional for users to have larger monitors usually 1080p for better clarity but any resolution will do.
- Better internet connections mean quicker outputs though even basic internet connections will also work just fine.

If these basic requirements are met, the application will run without issues or hassle. The larger monitor and faster connections are optional and are not necessarily required, though the experience will be slightly more refined.

3.1.3 Platforms and Tools Utilized

To build this application some platforms and tools were utilized. This application has front end, integration, back end and machine learning as well. Various Integrated Development Environments (IDE) were utilized to bring forth this application. Various languages, frameworks and libraries were utilized to make the application. Below all the details are given.

Integrated Development Environment (IDE):

- Google Colaboratory
- Visual Studio Code

Google Colaboratory was utilized to make the machine learning aspect of the application along with back end design and integration as well. Visual Studio Code was used for the front end design.

Programming Languages Used:

- Python
- HyperText Markup Language (HTML)
- Cascading Style Sheets (CSS)

For the machine learning, integration and back end Python language was utilized. For the front end HTML and CSS.

The following Python Libraries were used for this application.

- PyTorch
- Transformers
- GPT2-model

PyTorch and Transformers for the machine learning aspect and GPT2-model for the natural language processing aspect.

The Gradio framework was utilized to build the web application with added HTML and CSS. These are all the platforms and tools utilized to bring forth this project application.

3.2 Computational Requirements

To run the application on any device some basic computational requirements are necessary. Some of these may be optional. If the basic computational requirements are met then the application can run smoothly. The optional requirements ensure even smoother operation, though it is more of personal preference. The more the requirements are met, the more smoother and unique the operation of the application.

3.2.1 Hardware Requirements

For any device in particular any desktop or laptop some of the device requirements for running the application smoothly are given below in detail. Maximum ones are also mentioned though not necessary. The application will run smoothly if the minimum hardware requirements are met.

The hardware minimum and maximum requirements are given below in detail. This is for both desktops and laptops.

Table 3.1: Minimum and Maximum Hardware Requirements

Component	Minimum	Maximum
Processor (Intel / AMD)	Core i3 / Ryzen 3	Core i7 / Ryzen 7
Random Access Memory (RAM)	2GB	8GB
Graphics Memory	2GB	8GB
Operating System (OS)	Windows 7/8	Windows 10/11

All the minimum and maximum requirements for the application to run smoothly are given above. Any 4 or 6 core processor will ensure smooth operation, 8 core processor will mean quicker operations but not necessary. 2GB RAM is sufficient, though in larger workloads 8GB might be necessary. Application does not use much Graphics memory, 2GB is enough, though 8GB will be more than plenty. Might work on Windows 7 or 8, but for smoother experience Windows 10 or 11 is recommended.

3.2.2 Software Requirements

For the application to run some software requirements are necessary. These software will ensure smooth operation for the AI Thesis Helper. The required software are given below with necessary details.

Table 3.2: Software Requirements

Component	Software
Browser	Google Chrome or Mozilla Firefox
IDE	Google Colab and VScode
Languages	Python and HTML and CSS
Libraries	PyTorch, Transformers and GPT2-model

The browser is needed to run the application. The application is made in the IDE. The languages are used to build the AI Thesis Helper. The libraries are needed for their functions in the application. These are the software requirements for the application.

3.2.3 Other Requirements

The application will be more comfortable to use on desktop as opposed to laptop. Desktop provides more comfort and also due to the more powerful components will provide more smoother operations, though again it is the users choice and users personal preferences. The application will work on both devices well equally. This application is currently not supported on Mac or Linux or Android. In the future these updates may be available. For now only it is only available for desktops and laptops. Besides this, there is no other requirements.

3.3 Pros and Cons

The AI Thesis Helper comes with its own set of advantages and disadvantages. The advantages ensure this application is a grade above what is available. Though it does come with some disadvantages. Both the pros and cons of this project are discussed below.

3.3.1 Pros

The AI Thesis Helper comes with a set of advantages. The advantages are:

1. The application is very simple.
2. Any user can use it, not just students.
3. Gives answers quickly.
4. Answers can be changed.
5. Background color customization.
6. Browser based, no downloads necessary.

These are some of the pros of the AI Thesis Helper.

3.3.2 Cons

The AI Thesis Helper has some disadvantages. The disadvantages are:

1. Requires an internet connection.
2. Only works on some browsers.
3. Gives limited answers.
4. Works on Windows only.
5. Answers sometimes have mistakes.

These are some of the cons of the AI Thesis Helper.

3.4 Feasibility Report

The AI Thesis Helper comes with the given feasibility report. There is economic feasibility and technical feasibility. Only these two are taken as these are the ones the project mostly affects. Both are given below in detail.

3.4.1 Economic Feasibility

This project has some good economic feasibility. These are given below.

- Reduces time to finish thesis for both students and teachers.
- Less time needed to generate ideas.
- Less time needed for teachers to check or correct mistakes.
- Less stress for students and allows for more research time.
- One teacher can take more thesis students.

The AI Thesis Helper will allow these economic feasibility to come to fruition.

3.4.2 Technical Feasibility

The AI Thesis Helper comes with some technical feasibility as well. These are given below.

- Gives correct spellings and grammar automatically.
- Can generate answers relatively precisely.
- Answers given in small time.
- Thesis requires less resources.

These are some of the technical feasibility that comes with the project.

3.5 Model Diagrams

The use-case, class, data flow and entity relationship diagrams are given here to plan the project ahead. Use case shows user and system interactions, class diagrams shows different building blocks, data flow shows how data is transferred in the system and entity relationship shows relations among the data process and components.

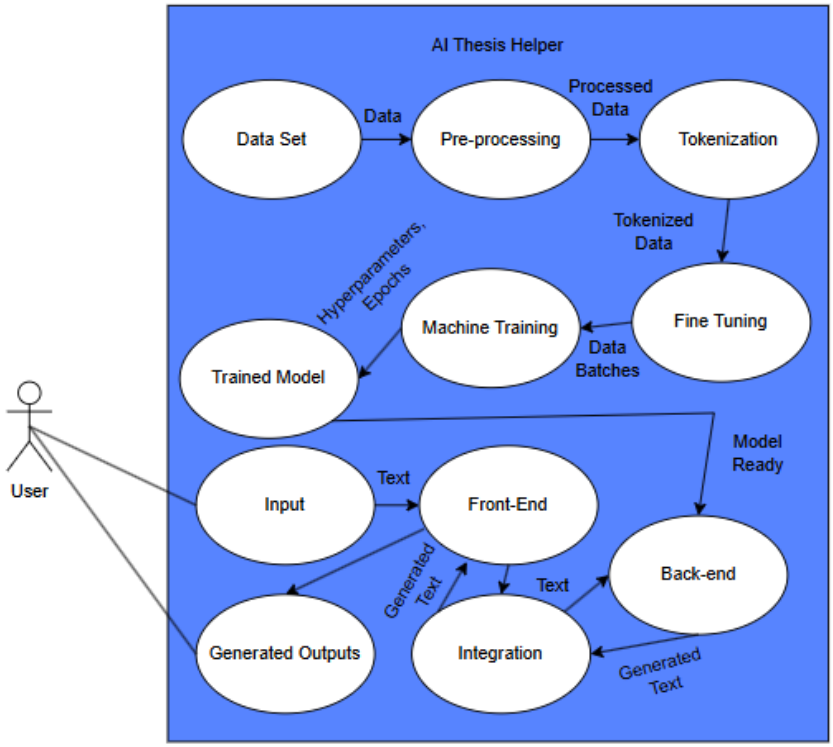


Figure 3.1: Use Case Diagram

As per Figure 3.1, user inputs text to system text goes to front end then to integration. integration takes text to back end and ready model generates text. Generated text is sent back to back end then to integration then to front end for user. Data is pre-processed, tokenized, fine tuned. Machine trained, saved and ready for use.

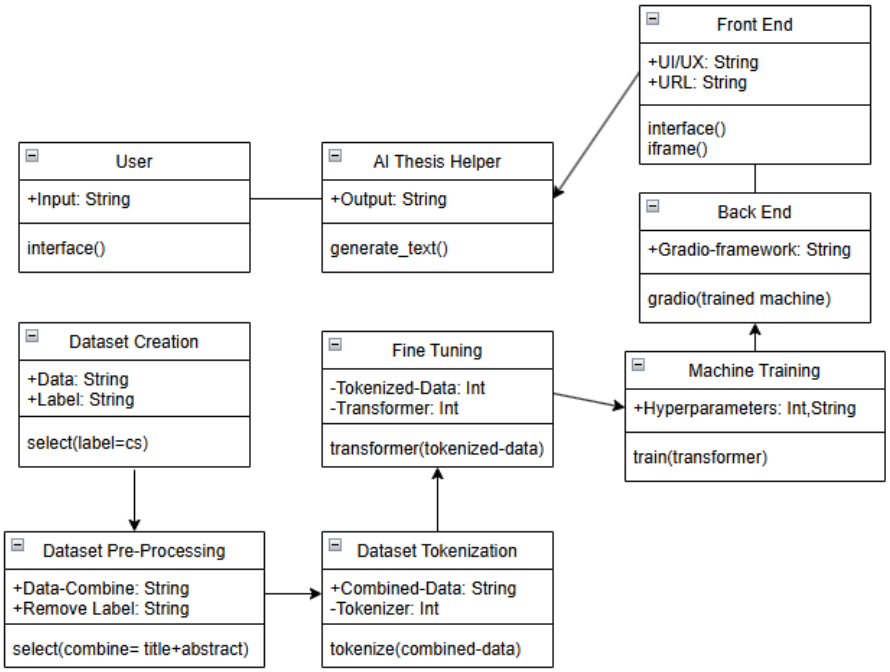


Figure 3.2: Class Diagram

As per Figure 3.2, user inputs text via interface, system generates output string.

Data creation makes dataset followed by pre-processing, tokenization, fine-tuning then machine training. Ready model saved to back-end and integrated to front-end.

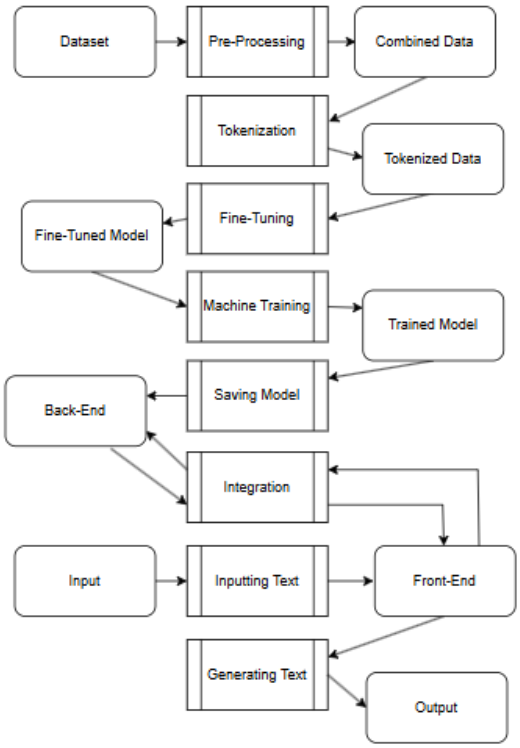


Figure 3.3: Data Flow Diagram

As per Figure 3.3, there are 8 processes involved all in the middle, right and left are external entities. Data flow given via arrows. Data flows both ways in integration.

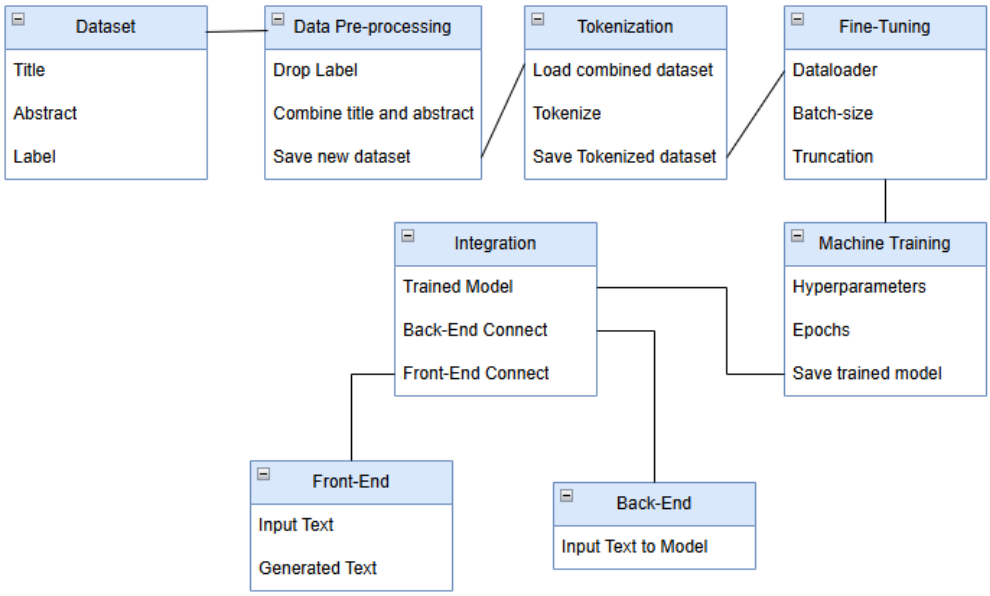


Figure 3.4: ER Diagram

As per Figure 3.4, very similar to class diagram again all components are given along with their relations between data transfers.

Chapter 4

Project Analysis

4.1 Project Workings

The project has been made using all the resources currently available. As such, the below sections talk about the dataset and all the details of the dataset as well. How the GPT model has been fine tuned with all details of the process. How the back-end and integration was made. How the front-end of the program was made using HTML and CSS. Along with the discussion of UI/UX components.

4.2 Dataset

For this specific task the dataset has been chosen carefully. This AI Thesis Helper for now is geared mostly towards computer science students though it can still answer other questions. As such most of the data it was fine tuned on is computer science papers from Arxiv. Here the Arxiv-100 dataset has been used. From the dataset, Arxiv-10-CS dataset has been made for this task. Below shows all the details.

4.2.1 Arxiv-100 Dataset and Arxiv-10-CS Dataset

The Arxiv-100 Dataset is a CSV file consisting of three columns - title, abstract and label. There are 100,000 papers of 10 different labels with 10,000 papers per label. All 10,000 papers taken with the label “cs” has been moved to a new csv file and has been named the Arxiv-10-CS dataset. The label column has been removed as all labels are “cs” so it is not needed.

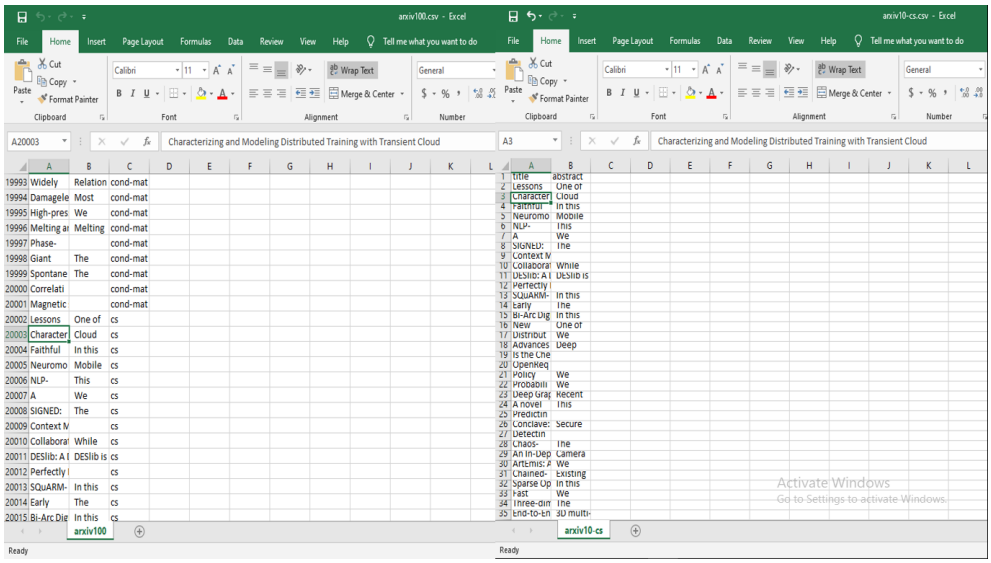


Figure 4.1: Arxiv-100 Dataset (left) and Arxiv-10-CS Dataset (Right)

As per Figure 4.1, the data with the label "cs" are shown. There are 10,000 papers with label, "cs" is computer science. The Arxiv10-CS dataset with two columns - title and abstract are shown.

4.2.2 Uploading Dataset to Google Colab

The Arxiv10-CS dataset requires some pre-processing before the dataset can be tokenized and then given into the GPT model.

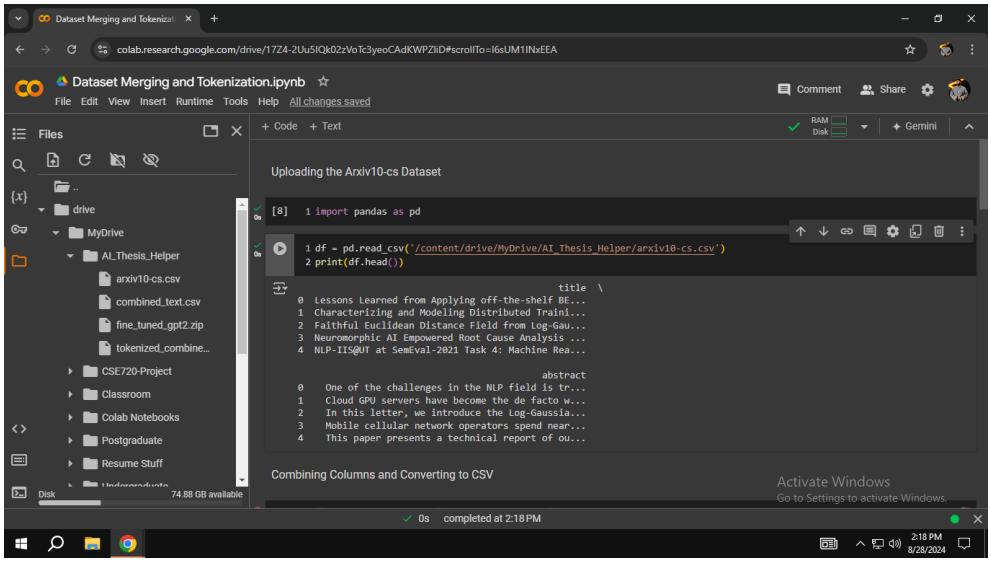


Figure 4.2: Dataset Uploaded to Colab

As per Figure 4.2, the dataset has been loaded to google drive, then the drive was mounted to colab. The path of the dataset has been given and read using pandas data-frame.

4.2.3 Dataset Pre-Processing

The title and abstract columns are then combined into the combined text column. The title and abstract columns are then dropped so only the combined text column remains. The dataset is converted to CSV format.

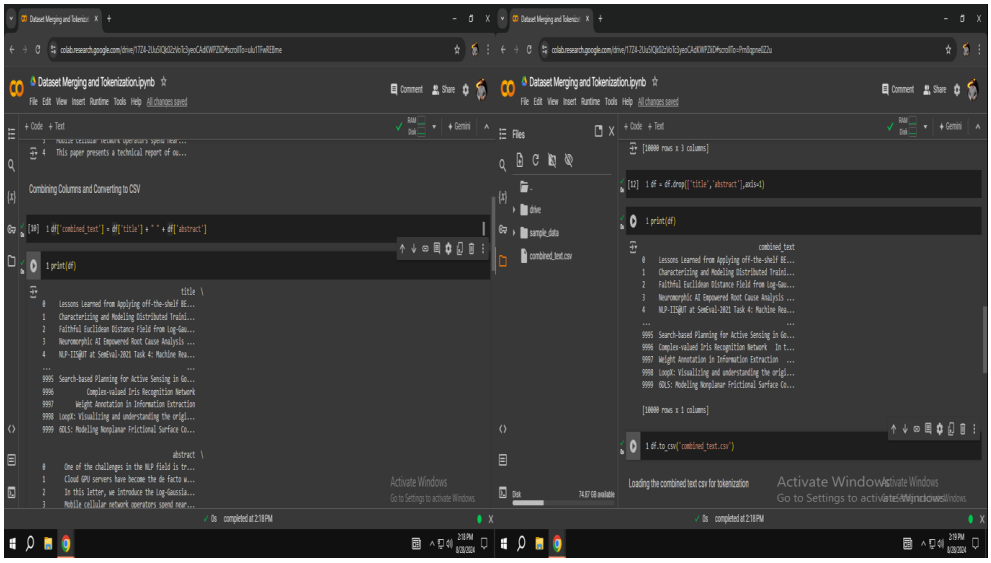


Figure 4.3: Dataset Column Combination, Removal and CSV conversion

4.2.4 Uploading Combined Dataset and Tokenization

The combined dataset CSV is then uploaded to drive and again using pandas dataframe it is accessed. GPT2Tokenizer is imported from Transformers package and a padding token is given. The tokenizer is applied to the combined text. Now the dataset is tokenized and has two columns combined text and tokenized text. Finally the combined-text column is dropped and dataset is converted to another CSV file.

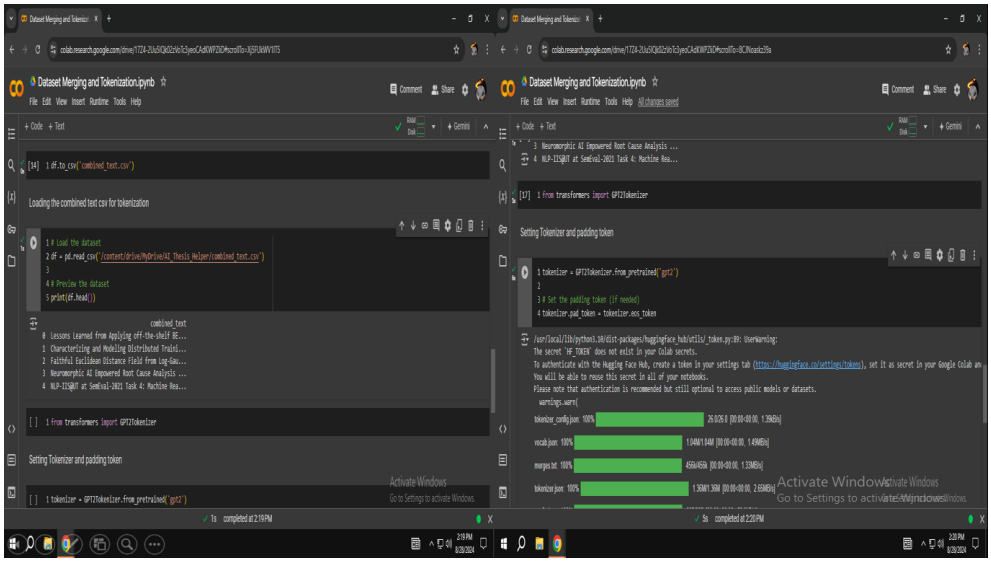


Figure 4.4: Combined Text Dataset Upload and GPT2 Import

As per Figure 4.4, GPT2Tokenizer imported. Tokenizer set with padded token.

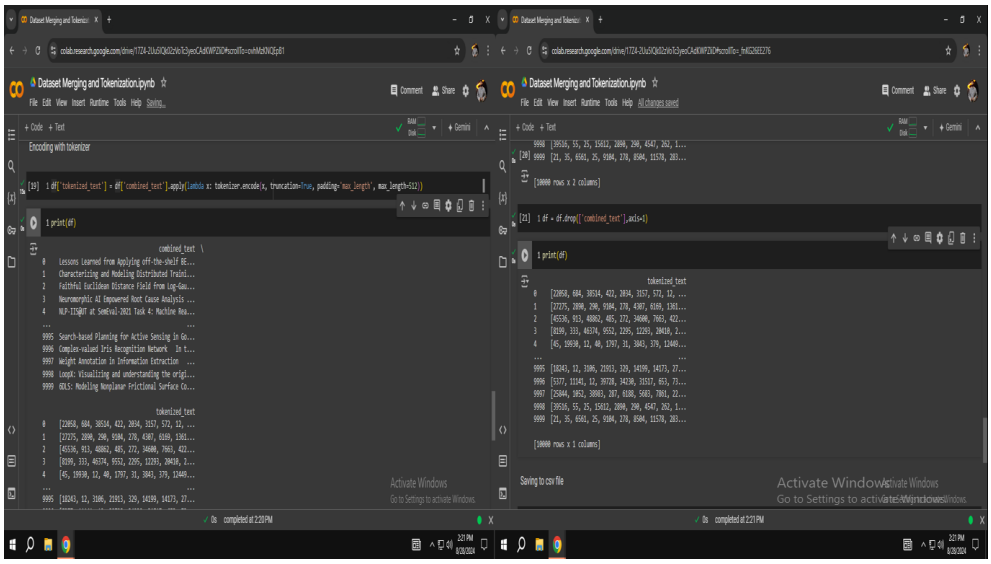


Figure 4.5: Tokenizer Applied on Combined Text Column and Column Removed

As per Figure 4.5, tokenizer is applied on the combined-text to make tokenizer-column. Tokenizer parameters- $truncation = true$, $padding = maxlength$ and $maxlength = 150$. Combined-text removed, only tokenized-text column remains. The tokenized dataset is then converted to CSV. This dataset will then be uploaded to another colab tab for fine tuning the GPT model.

4.3 Fine Tuning

Using the tokenized dataset, the GPT can now be fine tuned to answer computer science related thesis questions. The hyperparameters are set and then the training loop runs. In this case colab runtime is connected to a back-end T4 GPU. The trained model is saved and zipped for download with the size of the zip file shown.

4.3.1 Uploading Tokenized Dataset And GPT-2 Dataset

Here the tokenized dataset is uploaded from the mounted google drive. Then the GPT-2 Dataset class is made with three functions - initialize, length and get-item.

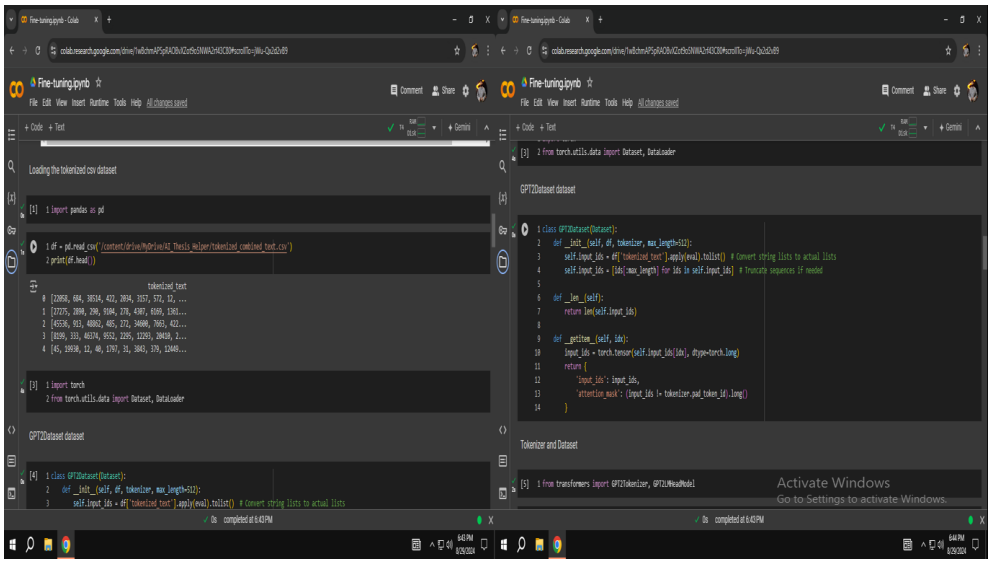


Figure 4.6: Tokenized Dataset Uploaded and GPT2 Dataset Class Made

As per Figure 4.6, GPT2Dataset class is made with three functions. *Initialize* starts the sequence, the *length* is taken of dataset and *getitem* fetches the items.

4.3.2 Tokenizer, Batching and Loading Pre-Trained Model

GPT2Tokenizer and GPT2LMHeadModel are imported from transformers package with a padding token set. Dataset and Dataloader variables are set for batching. The pre-trained model is then loaded.

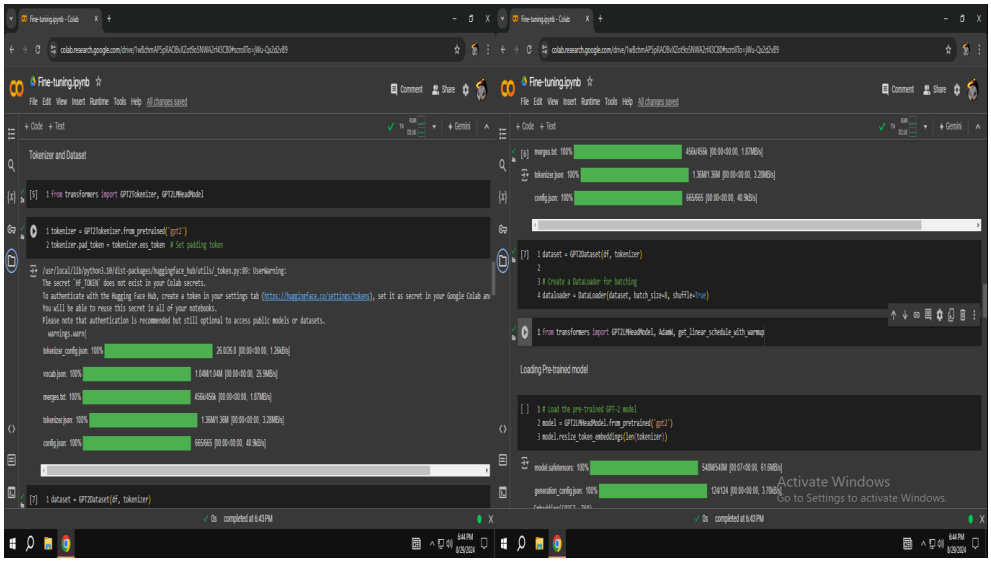


Figure 4.7: Importing then Setting Tokenizer, Dataset and Dataloader

As per Figure 4.7, tokenizer is imported and set with a padding token. Dataloader parameters for making batches are $batch - size = 8$ and $shuffle = True$.

4.3.3 Model Details, Hyperparameters and Training

The model is shown with all details, with the hyperparameters used for training. Resource usage is monitored and finally the machine is trained in a training loop.

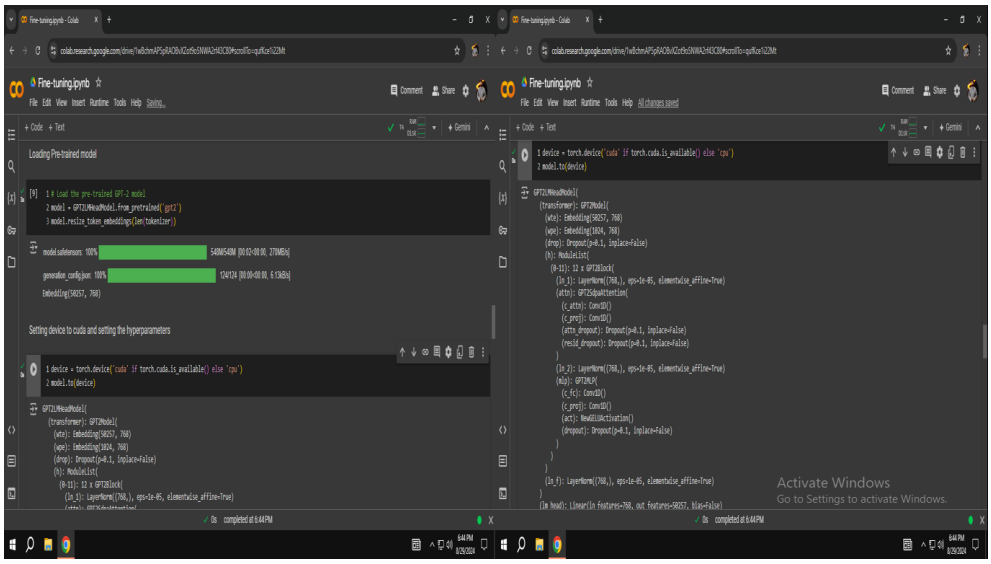


Figure 4.8: Loading Pre-Trained Model and Model Details

As per Figure 4.8, the pre-trained model is loaded and token embedding is resized as per the length of the tokenizer.

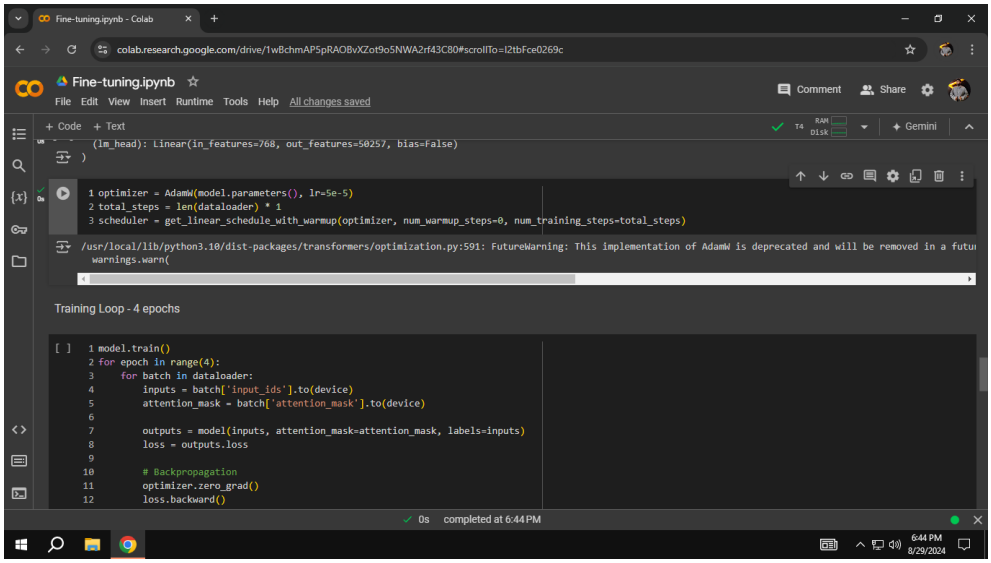


Figure 4.9: The Hyperparameters for Training

As per Figure 4.9, here the hyperparameters are set. Optimizer is AdamW and the learning rate is $lr = 5e^{-5}$. Total steps is length of data loader which is $10,000 * 1 = 10,000$. Scheduler is given with the necessary parameters.

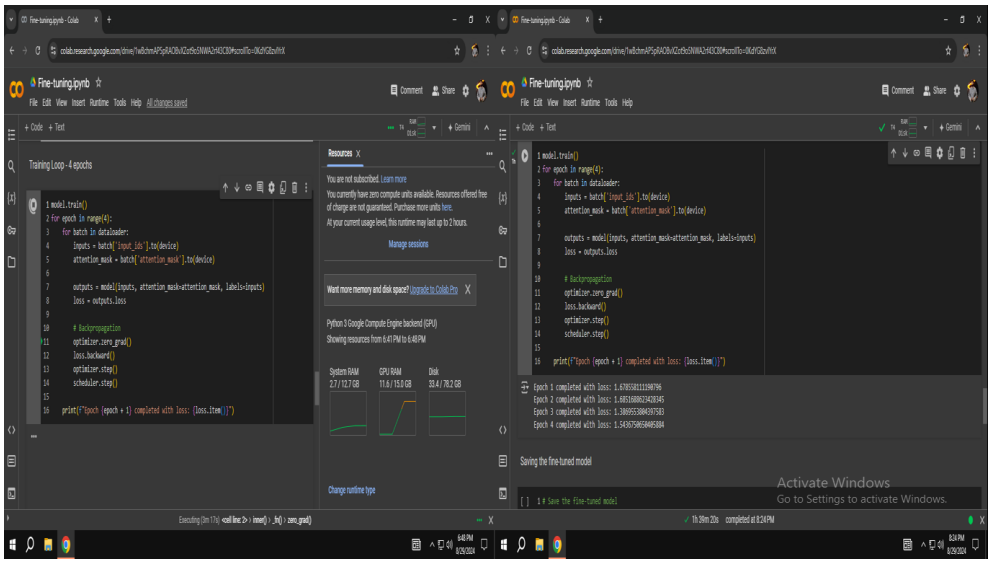


Figure 4.10: Resource Usage and Training Loop

As per Figure 4.10, GPT-2 model often takes a large amount of resources to train even for a small number of epochs. Since it is already pre-trained, a small number of epochs is needed for fine-tuning. GPT-2 requires about 11.6GB of GPU-RAM of the available 15GB GPU-RAM of T4 GPU. As such, excessive training is not possible and will be beyond colab GPU-limits. The machine is trained for 4 epochs. The losses are recorded in each epoch. As epochs increase, the losses decrease up to the third epoch, in the fourth one it increases a bit due to randomness. Total training time for machine is 1 hour, 39 minutes and 20 seconds. Each epoch took roughly about 25 minutes. 4 epochs are enough to fine-tune the pre-trained machine.

4.3.4 Saving Model, Testing and Downloading Saved Model

Here the model is saved, then a small test is taken to see a sample of the performance and then model is zipped and downloaded.

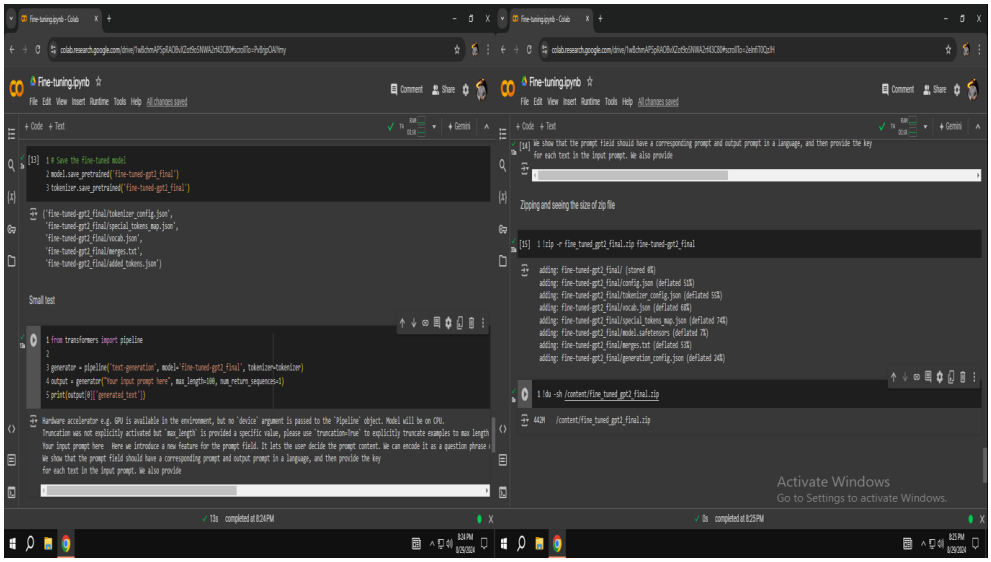


Figure 4.11: Saving Model and Zipping

As per Figure 4.11, here the fine-tuned model is saved. And a small test is run to peek at the performance. The model is being zipped and the size is checked. The size of the model is 442 megabytes.

4.4 Back End

The back-end of this project is done in colab. Gradio is used for the interface framework. The zipped model is unzipped. Gradio and other imports from transformers package are imported, as per Figure 4.12. Generator is made with fine-tuned model. Text-Generation function is made, the back-end interface is made and launched. Gradio is installed in the colab environment.

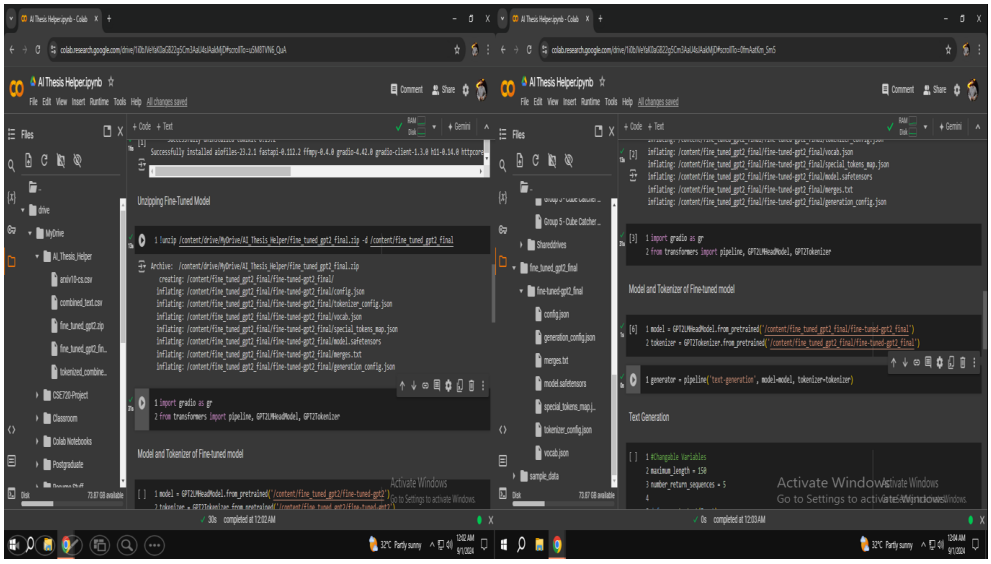


Figure 4.12: Unzipping Fine-tuned Model, Importing Items and Setting Generator

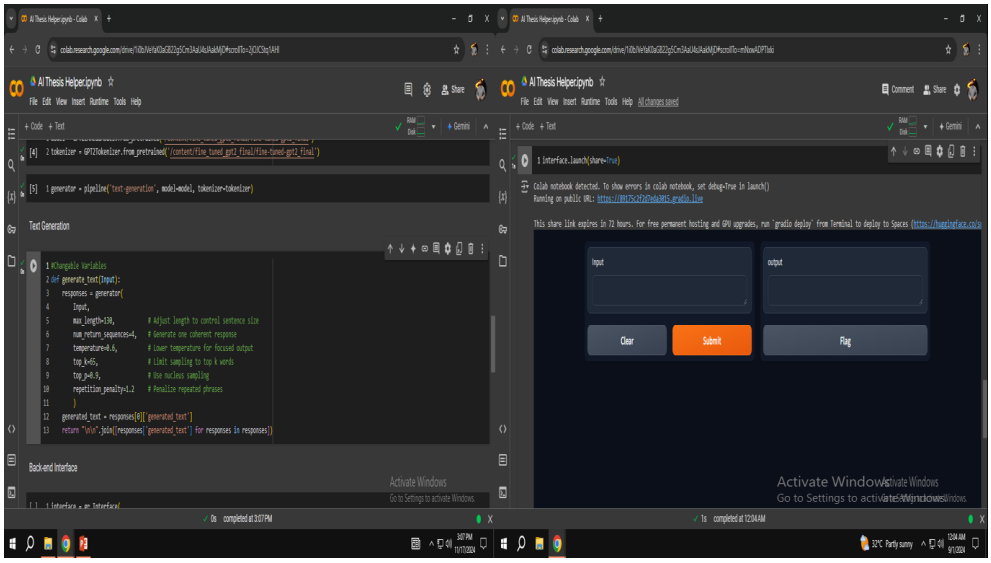


Figure 4.13: Text Generation Function, Back-end Interface and Launch

As per Figure 4.13, text-generation function is made with multiple changeable variables. Higher numbers mean more processing. Gradio interface is launched with the

share parameter as true. A public URL is generated along with the interface which is active for 3 days.

4.5 Integration

The front-end and back-end are now integrated together. Gradio interface is hosted on a public URL. It is generated by Gradio. This shareable URL is then copied and pasted to the src of iframe in front-end HTML while iframe details are given in CSS.

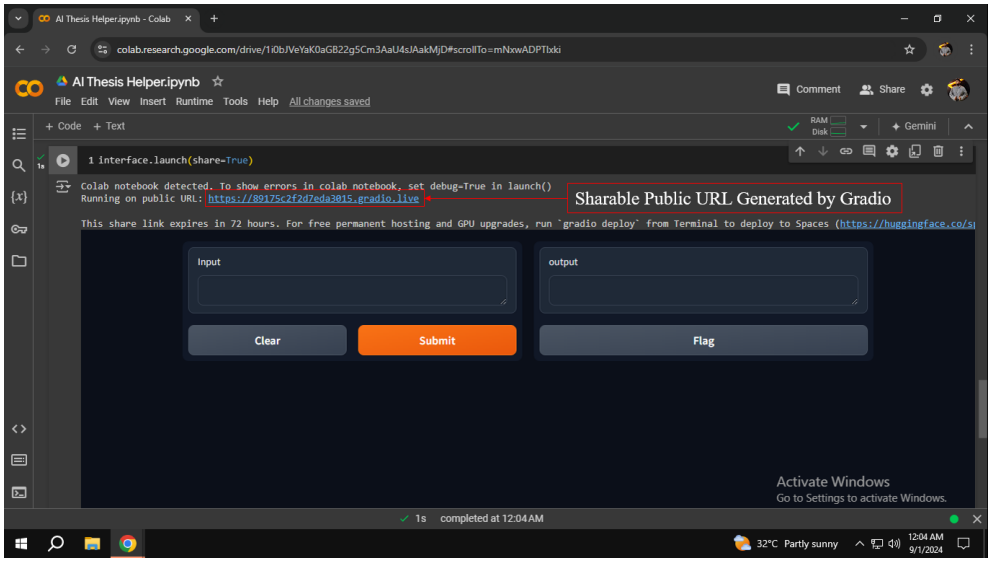


Figure 4.14: Gradio Generated Shareable Public URL in Colab

As per Figure 4.14, Gradio interface generates a shareable public URL, this URL is copy and pasted into the Front-end HTML.

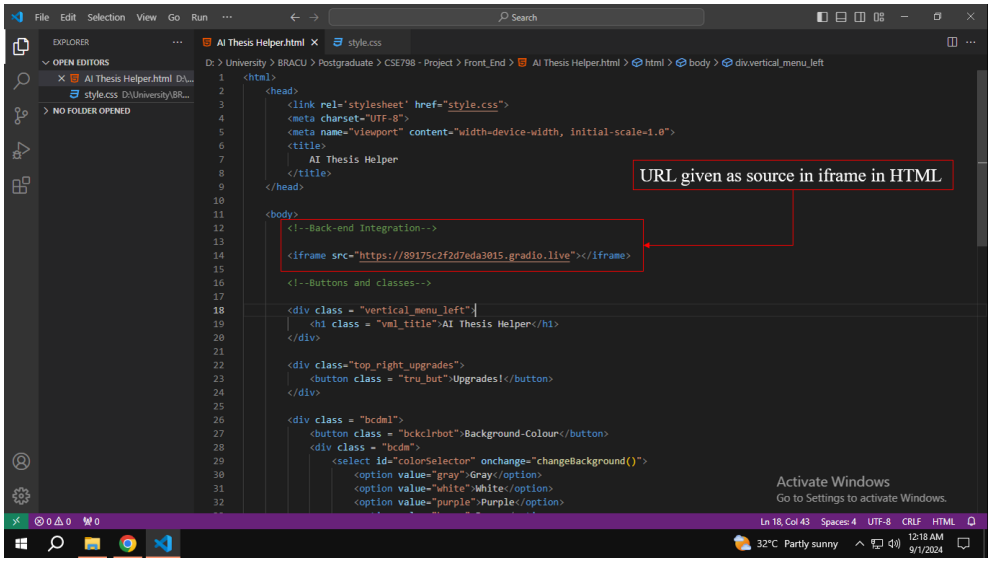


Figure 4.15: URL Pasted into Front-end HTML

As per Figure 4.15, in the front-end HTML iframe component the source - *src* is given the pasted URL.

4.6 Front End

The front-end was made with HTML and CSS in VScode. According to Alegado et al., visual style of graphical user interface is interface design. Should guarantee easy access to most features [10]. As per Fadhil, chatbot should have a close to human personality in giving answers to make it more appealing [2]. Simple design used.

4.6.1 HTML

Figure 4.16 shows all the HTML.

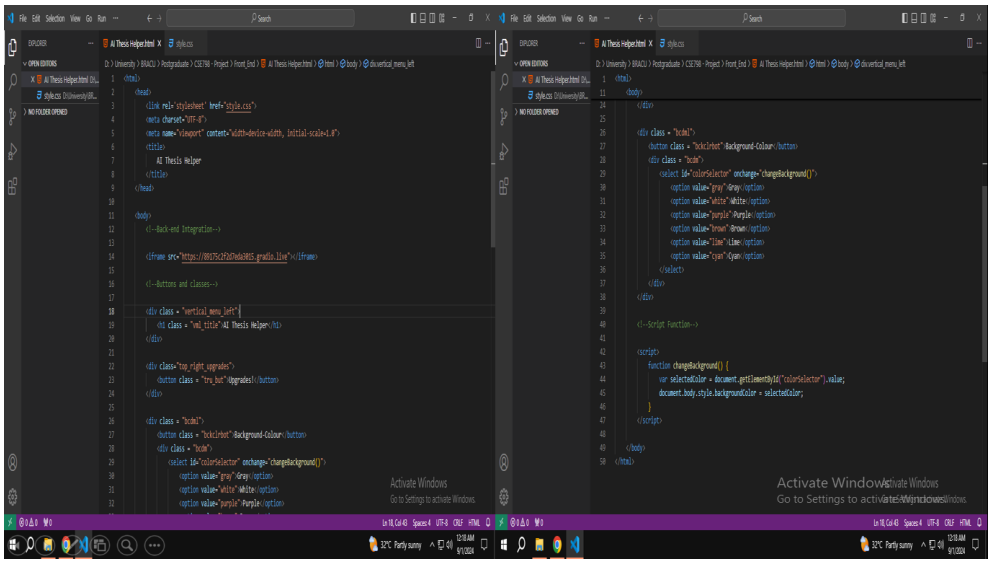


Figure 4.16: All HTML

4.6.2 CSS

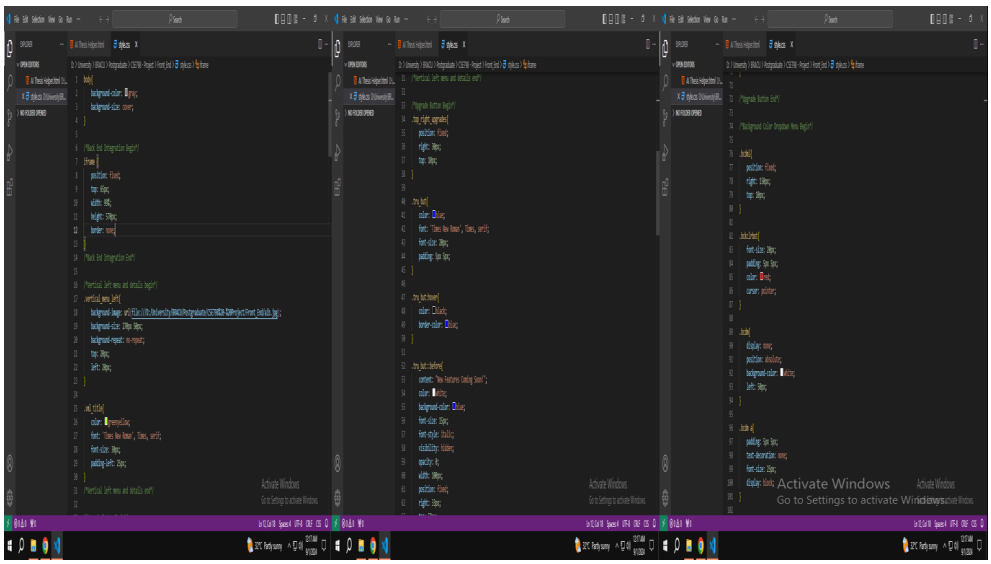


Figure 4.17: All CSS

Figure 4.17 shows all the CSS.

4.7 UI/UX Components

The AI Thesis Helper has a set of UI/UX components that make up the website. According to B, Lumingkewas and Rofi'i, user centered design is utilized to update UI/UX components to meet the required criteria [15]. As per Duijst, the modern concept of user experience is consistent of users emotions, psychological responses and behaviors. Using the product should leave users feeling very good [1]. Below shows a labeled image of the UI/UX components.

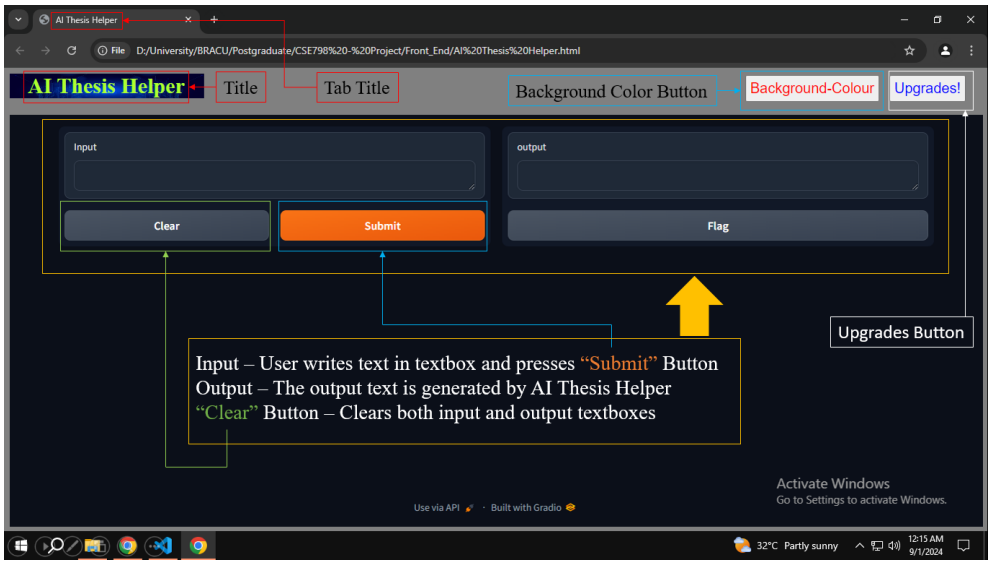


Figure 4.18: All Components Labeled

4.7.1 Input and Output Text

The input and output text functionalities are shown. User inputs text, submits then AI thesis Helper generates output. Clear button clears both input and output.

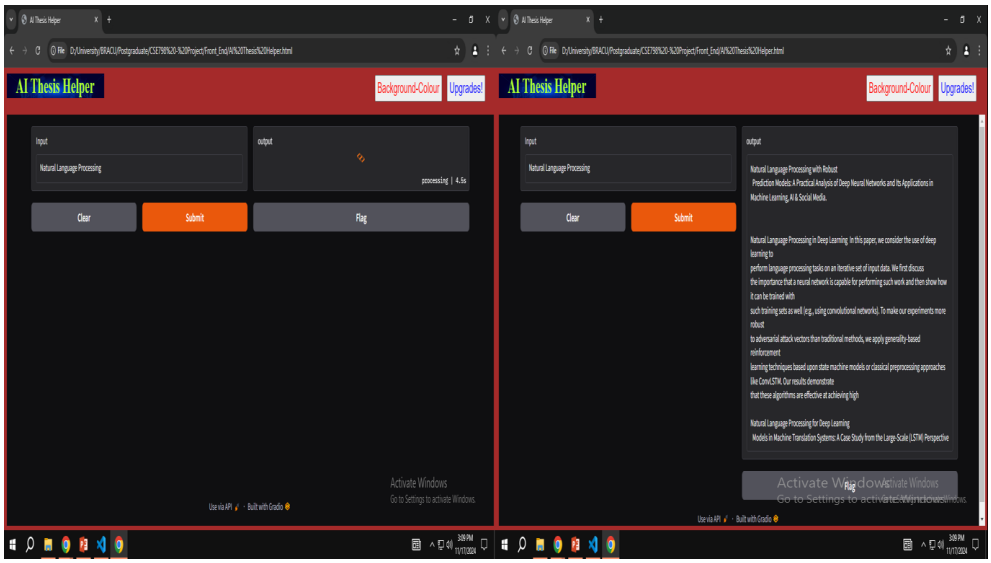


Figure 4.19: Processing and Generated Output

4.7.2 Background Color Button

The background color button is used to change the background color of the AI Thesis Helper. It does not change the color of the input output text background but the background behind it.

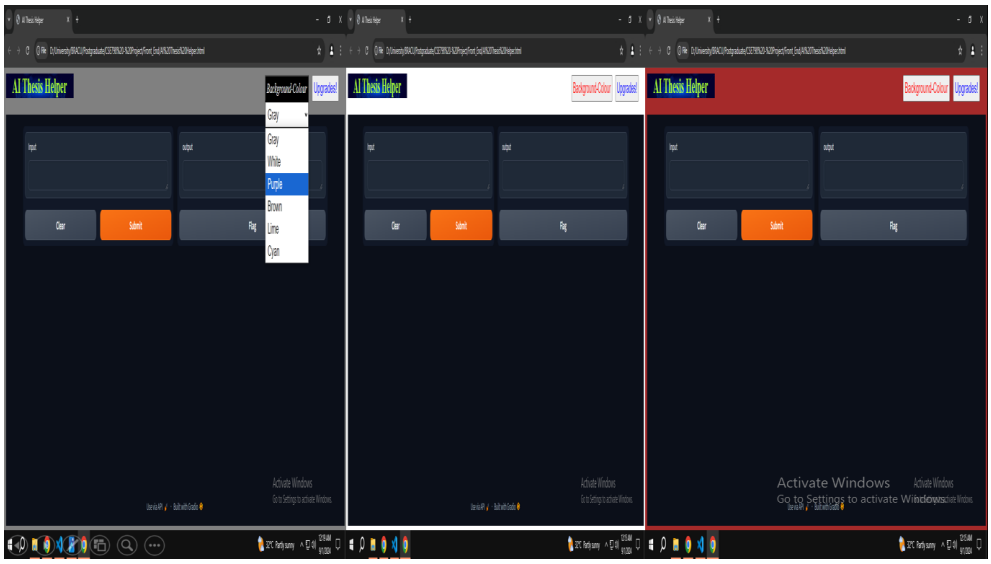


Figure 4.20: Background Colours Menu and Color Changes - White, Brown

4.7.3 Upgrades Button

The upgrades button is there to show that new features will come soon in the future. It just displays a small message.

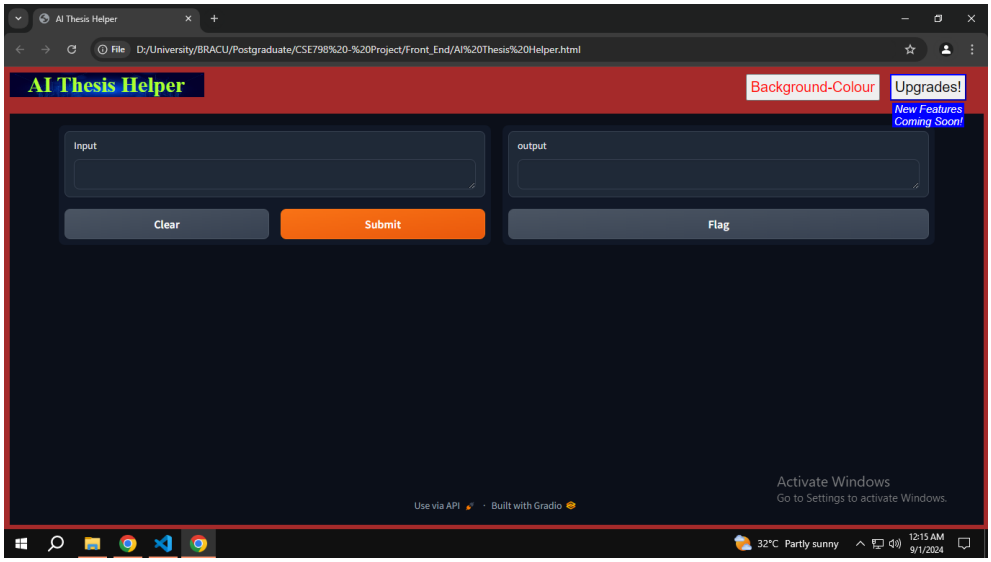


Figure 4.21: Upgrades Button Showing Message

Chapter 5

Implementation

5.1 Project Implementation

Here the generated answers of the chatbot are showcased along with the accuracy and the perplexity of the answers. Finally, the answer comparisons between this model and other current chatbots. Spelling, grammer and writing quality is accessed as well. According to Kumar and Mahmoud, ease of answering, information flow and semantic coherence are the three rewards of a good implemented chatbot system [8]. As per Skuridin and Wynn, the most challenging part of any chatbot is implementation. Sometimes users may not like new ways of communicating and chatbots may give erroneous answers [31]. According to Lalwani et al., it is not possible to get all data in a single interface without multiple components or many windows. Simplicity ensures enjoyment of the product [4].

5.2 Chat-bot Answers

Here, some chatbot answers are shown to some common topic questions. At first, it answers computer science based questions, then an electrical engineering based question is asked to see how it handles questions beyond computer science.

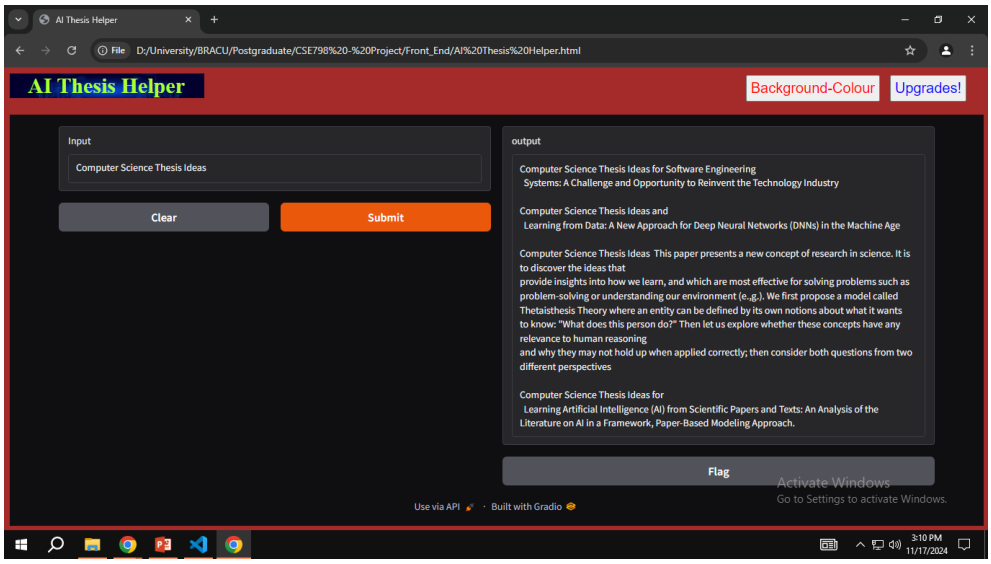


Figure 5.1: Text Input - "Computer Science Thesis Ideas"

As per Figure 5.1, AI Thesis Helper generates answer to the given text above. Answers have clear spacing, indicating different papers. Also stated the paper with title and abstract. 4 most relevant papers are given as answer on topic. This is computer science related.

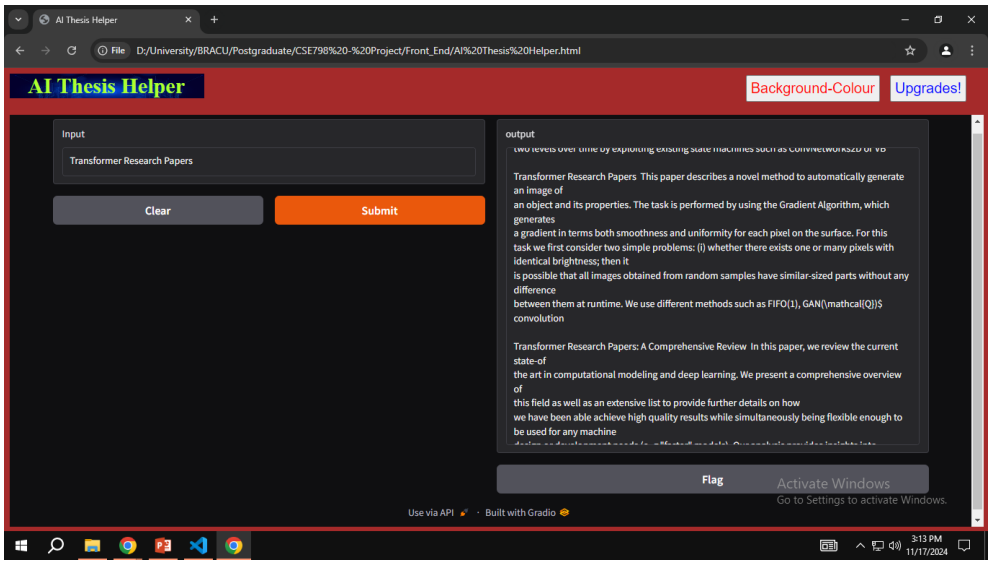


Figure 5.2: Text Input - "Transformer Research Papers"

As per Figure 5.2, again generates another answer with 4 papers that are relevant to the topic stating the title and abstract. Large spaces between indicate different papers. Again, computer science related.

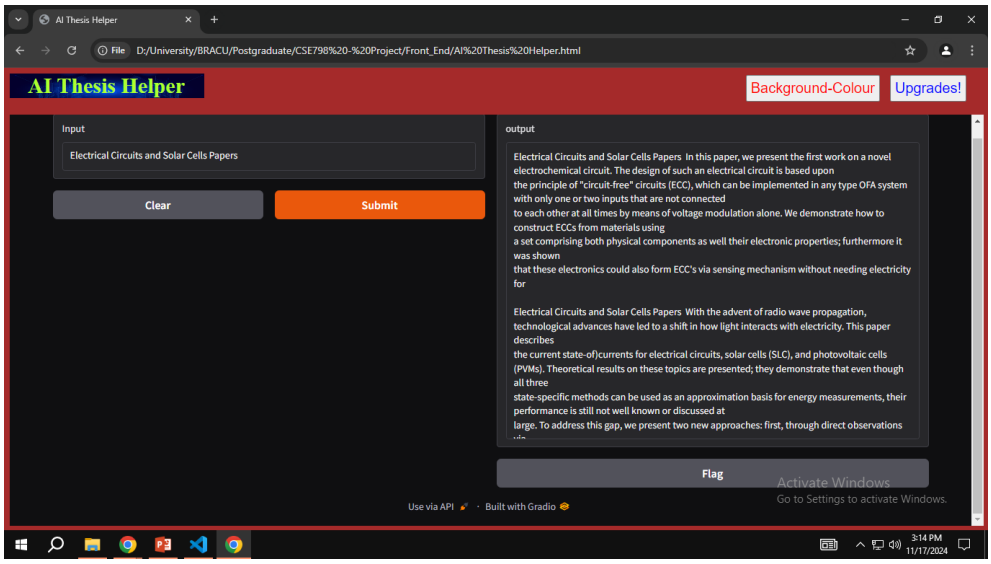


Figure 5.3: Text Input - "Electrical Circuits and Solar Cells Papers"

As per Figure 5.3, the question is electrical and electronic engineering related, yet chatbot gives a good answer. Some computer science papers coincide with electronic engineering papers and also model is pre-trained. Again, 4 papers with title and abstracts relevant to the topic with spacing.

Chatbot generates relatively good output texts for the input texts. Overall, the answers are concise, have enough information and are very useful.

5.3 Model Evaluations

Now the chatbot precision, recall, F1 score, perplexity and grammar, spelling accuracy are calculated. The loss during training of the fine-tuned model is also considered. According to Kumar, Kuo and Paliwoda, calculating accuracy and usefulness of any chatbot on any peculiar subject determines how efficient the chatbot is. The statements must make sense, be concise and be readable [25].

5.3.1 Model Loss During Training

The model is fine-tuned during the training phase. The following hyperparameters are utilized. Optimizer is AdamW and the learning rate is $lr = 5e^{-}5$. Total steps is length of data loader which is 10,000. Scheduler is given with the necessary parameters. The model runs for 4 epochs. GPT2 is already pre-trained so not many epochs is necessary. Total time take is 1 hour, 39 minutes and 20 seconds. Each epoch took roughly about 25 minutes. The T4 GPU in colab is used for this.

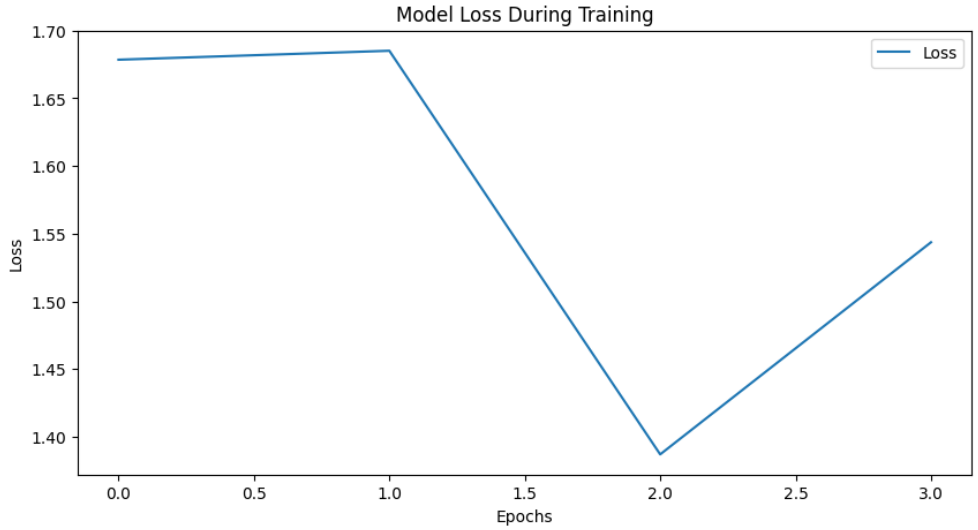


Figure 5.4: Loss During Training

As per Figure 5.4, from Epochs 0-2 the loss decreases, but on Epoch 3 loss increased slightly due to random samples. This is overall a very good model as the loss is small in nature.

5.3.2 Precision, Recall and F1 Score

The model's precision, recall and F1 score are calculated to evaluate the performance. In this case the BERTScore is utilized to see the precision, recall and F1 score of the model's generated text. The same text is given to ChatGPT and the generated output is used as a reference text to see how well the model fares. According to Zhang, Kishore, Wu and Weinberger, BERTScore is an automatic evaluation metric. Similarity using BERT embeddings is computed instead of finding exact matches. Correlates better than existing metrics with human judgments [6].

Table 5.1: Precision, Recall and F1 Score

Text Input	Precision(%)	Recall(%)	F1 Score(%)
Computer Science Thesis Ideas	79.5	79.5	79.5
Transformer Research Papers	79.7	78.4	79.1
Electrical Circuits and Solar Cells Papers	80.2	78.1	79.1
Total Average	79.8	78.7	79.2

As per Table 5.1, the model was able to give very good answers and was able to clearly provide information both within and outside of its fine-tuning. It shows answers that are relatively close to what ChatGPT would generate. Overall precision, recall and F1 score are all close to about 80%. So the model is very good at text generation.

5.3.3 Perplexity

The model perplexity is shown in a tabular format. According to Kolena, Bai, Hart and Chen, perplexity is basically the probability distribution of words generated by

the model. Quantifies how well the model can predict the next word in any given sequence. It is the exponential of a negative log-likelihood of a sequence. The lower the perplexity, the better the model.

$$Perplexity = exp(-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{context})) \tag{5.1}$$

Where x_i - generated token, and $x_{context}$ - preceding tokens that current generated token is conditioned on [34].

Table 5.2: Perplexity

Text Input	Perplexity
Computer Science Thesis Ideas	16.588
Transformer Research Papers	14.426
Electrical Circuits and Solar Cells Papers	14.134
Total Average	15.049

As per Table 5.2, the first text input shows relatively low perplexity, the rest two texts show even lower perplexity, overall perplexity is 15. Perplexity levels less than 20 are considered to be very accurate. So the model is very accurate. Perplexity reading below 20 means that the chat-bot does not suffer from much hallucinations. Hallucinations happen when AI gets data and tries to make up text that makes no sense. There can be some irregular sentences but AI does not suffer that much from hallucinations.

5.3.4 Spelling and Grammar Accuracy

To check the spelling and grammar accuracy of the AI Thesis helper in comparison to ChatGPT and Perplexity AI, three online checkers are utilized to measure the performance. QuillBot is used to access over all writing quality, Scribber for grammatical errors and Zoho Writer for spellings. In all three chat bots the text "Machine Learning" is given as input. The generated outputs are then pasted into the checkers and scores are taken.

Table 5.3: Caption

Checker	AI Thesis Helper	ChatGPT	Perplexity AI
Spelling - Zoho Writer	0.04	0.06	0.02
Grammar - Scribber	0.15	0.20	0.10
Overall - QuillBot	0.72	0.77	0.78

As per Table 5.3, spell check is done by Zoho Writer, an online spell checking platform. All measurements are in percentage. AI Thesis Helper has 4 mistakes, ChatGPT 6 mistakes and Perplexity AI 2 mistakes. Grammar check is done by Scribber, an online grammar checking platform. Counts the grammatical errors. AI Thesis Helper has 15 errors, ChatGPT 20 errors and Perplexity AI 10 errors. Lastly, Overall writing quality is checked by QuillBot, and online writing analysis tool - similar to Grammerley. QuillBot accesses the writing quality and gives a

percentage out of 100. AI Thesis Helper has a score of 72%, ChatGPT 77% and Perplexity AI 78%. Overall, AI Thesis helper is fairly close to the other chatbots in spelling, grammar and writing quality.

5.4 Answer Comparisons

Now for some comparisons of answers from different available chatbot AI. According to Fitria, various different chatbot AI, answer the same questions differently. It can be due to the structure of the machine, which words have more preference, different tokenization techniques etc [23]. In this case, the AI Thesis Helper's generated answer to the input text - "Computer Science Thesis Ideas" is compared to the generated answers of ChatGPT and Perplexity AI.

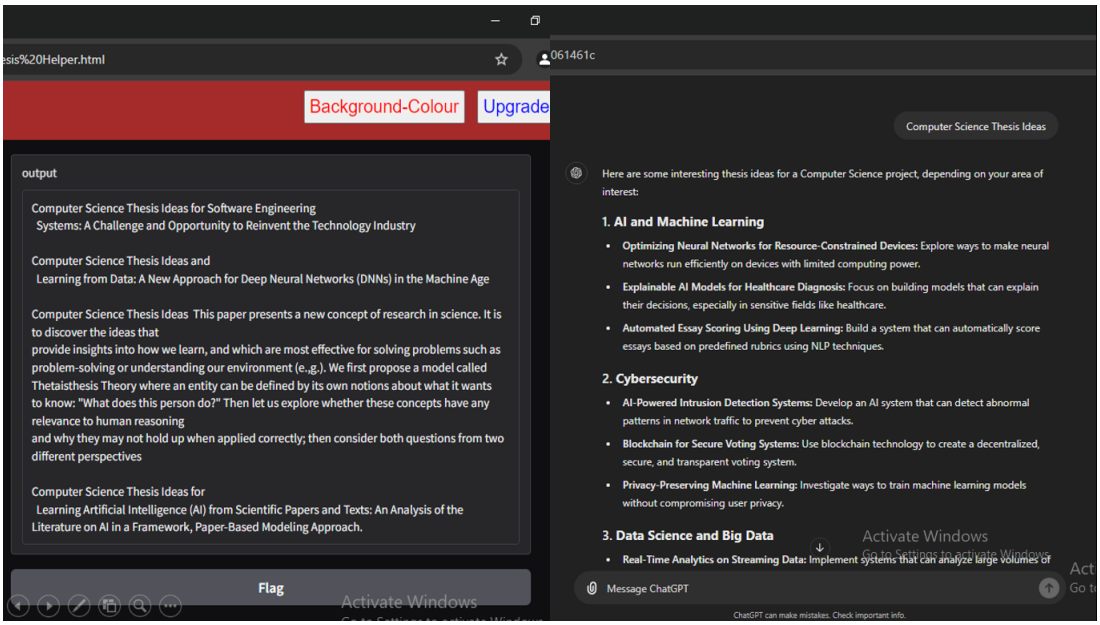


Figure 5.5: AI Thesis Helper (Left) and ChatGPT (Right)

As per Figure 5.5, AI Thesis Helper and ChatGPT have somewhat common answers, albeit ChatGPT's answer is more fine-tuned and more easier to read. AI Thesis Helper's answer is also similar, though more so in a continuous sequence. Also ChatGPT uses the updated GPT-4 which is even more powerful. Nonetheless, the answers are somewhat similar, though ChatGPT is more easier to read.

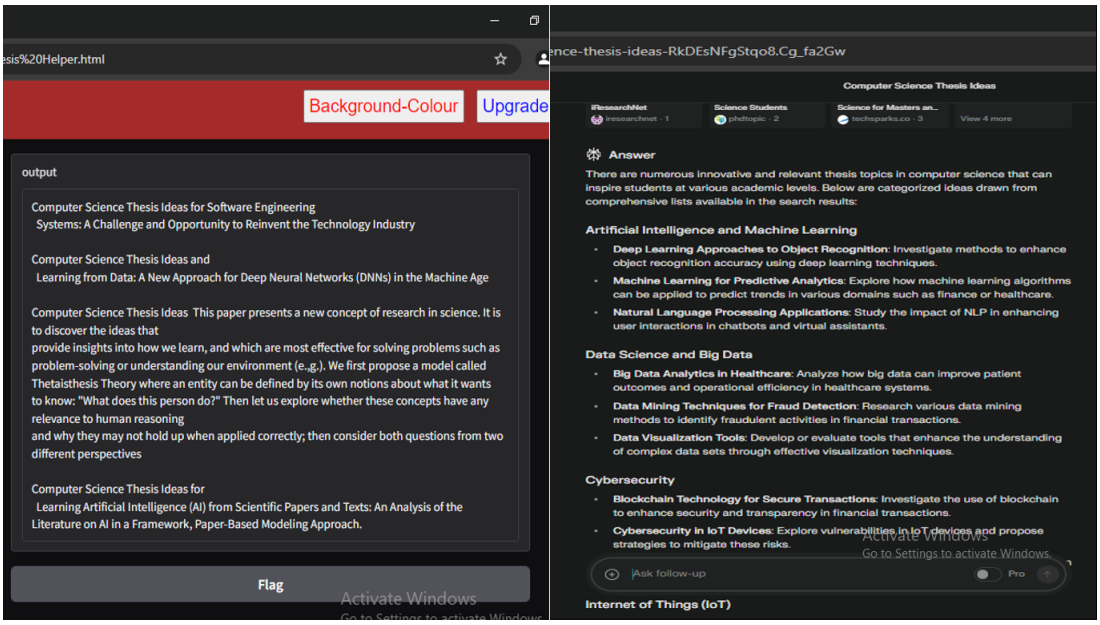


Figure 5.6: AI Thesis Helper (Left) and Perplexity AI (Right)

As per Figure 5.6, AI Thesis Helper and Perplexity AI have also similar answers, though again Perplexity AI is more fine-tuned and easier to read. Again, the AI Thesis Helper's answers are also similar, but appear more continuous. Again, Perplexity AI also uses up to date advanced GPT's. Nevertheless, both answers are rather similar, albeit Perplexity AI's answers are easier to read.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In conclusion, to help the students with their thesis the AI thesis helper was made. The motivation delves in the reasons. To increase the functionality of AI chatbots and to solve a specific problem. The problem was discussed with the project aims, objectives and challenges. The aim of this project was to help both students and teachers. The objectives of the project - to make spelling and grammar checking, to summarize and to generate good topics easier. The challenges were resource limits and dataset availability. The literature review discusses large language models and the structures and inner workings are explained. Transformers are explained in great detail. ChatGPT is discussed. Other models are discussed and why GPT was chosen over others, also due to GPT being a decoder model. The requirement analysis for both user and system are elaborated, along with all the tools used to make the application. The IDEs - Google Colab and Visual Studio Code. The languages - Python, HTML and CSS. The libraries - PyTorch, Transformers and GPT2-model. Computational requirements both hardware and software along with other optional ones are elaborated. The pros and cons are briefly discussed along with the feasibility reports both economic and technical. The model diagrams are given and discussed as well. The project analysis is discussed. First, the dataset is made then data pre-processing is carried out to combine the text and then tokenize it. Secondly, the tokenized dataset is used to fine-tune the model. The training is carried out in the training loop with given hyperparameters. The model is then saved, zipped and used. Thirdly the back-end, integration and front-end with UI/UX components are explained in full detail with all workings along with Gradio framework. Finally, Project implementation discusses the chatbot generated answers. The accuracy and perplexity of the answers are calculated and displayed in a tabular format, along with the model loss during training. The model shows precision, recall and F1 score of about 80% and a perplexity score of 15 which is below 20 and shows good text generation. It also means the AI does not suffer much from hallucination. Spelling, grammar and writing quality is also accessed in comparison to ChatGPT and Perplexity AI. Zoho Writer, Scribber and QuillBot - three online checkers are utilized. A comparison of two other chatbots - answering the same questions is also made to assess the performance of the model. Future work will improve the model and add more features.

6.2 Future Work

In the future, this project will receive significant upgrades. Currently this project only runs on either Google Chrome and Mozilla FireFox. It will be designed to run on others browsers like OperaGX, BraveBrowser, Safari and many linux based browsers as well. The current project is only designed for desktops and laptops - soon it will become available on smartphones, tablets and notepads as well. Currently the system only runs on Windows 10 or 11. May also run on Windows 8 or 7. The upgrades will allow this application to run on MacOS and various linux distributions like Ubuntu, Fedora, LinuxMint, Gentoo and Arch etc. Also there will be reverse compatible mode which will allow it to run on previous versions of Windows very smoothly - like Windows XP or 2000 etc. The application will also have various customization added to it. Letter size changing, color palette, different fonts and more background changes. Different modes will also be added to the project so the chatbot can be more refined for any particular task or to receive very narrow and specific answers. More models for the chatbot will be available as well. Chatbot may also have higher level features added at a much later time. Enhanced coding generation and equation capabilities are some of the higher functions that will be added. Chatbot can also be implemented on wireless devices and may act as remote tutors for students. There is a very promising outlook for this project and many Natural Language Processing (NLP) related chatbot projects in the future.

Bibliography

[1] D. Duijst, "Can we improve the user experience of chatbots with personalisation?," Jul. 2017. DOI: 10.13140/RG.2.2.36112.92165.

[2] A. Fadhil, "Domain specific design patterns: Designing for conversational user interfaces," *CoRR*, vol. abs/1802.09055, 2018. arXiv: 1802.09055. [Online]. Available: http://arxiv.org/abs/1802.09055.

[3] P. John and N. Woll, "Using grammar checkers in the esl classroom: The adequacy of automatic corrective feedback," pp. 118–123, Dec. 2018. DOI: 10.14705/rpnet.2018.26.823.

[4] V. R. Tarun Lalwani, "Implementation-of-a-chatbot-system-using-ai-and-nlp," *International Journal of Innovative Research in Computer Science and Technology (IJIRCST)*, vol. 6, no. 3, pp. 26–30, 2018, ISSN: 2347 - 5552. DOI: 10.21276/ijircst.2018.6.3.2.

[5] J. Park, "An ai-based english grammar checker vs. human raters in evaluating efl learners' writing," vol. 22, pp. 112–131. Sep. 2019. DOI: 10.15702/mall.2019.22.1.112.

[6] T. Zhang, V. Kishore, F. Wu, K. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," Apr. 2019.

[7] D. Hládek, J. Staš, and M. Pleva, "Survey of automatic spelling correction," *Electronics*, vol. 9, no. 10, 2020, ISSN: 2079-9292. DOI: 10.3390/electronics9101670. [Online]. Available: https://www.mdpi.com/2079-9292/9/10/1670.

[8] R. Kumar and M. Mahmoud, "A review on chatbot design and implementation techniques," *International Research Journal of Engineering Science Technology and Innovation*, vol. 07, p. 2791, Feb. 2020.

[9] R. Tamrakar and N. Wani, "Design and development of chatbot: A review," Apr. 2021.

[10] R. Alegado and R. Luciano, "Design and development of chatbot request system," Jun. 2022. DOI: 10.13140/RG.2.2.11593.01123.

[11] M. Hämäläinen, K. Alnajjar, and T. Poibeau, "Modern french poetry generation with roberta and gpt-2," Jun. 2022.

[12] W. Lun, M. Muhammad, K. P. Chuah, N. Saimi, A. Maarop, and R. Elias, "Did you run the telegram? use of mobile spelling checker on academic writing," *Multilingual Academic Journal of Education and Social Sciences*, vol. 10, pp. 1–19, Jan. 2022. DOI: 10.46886/MAJESS/v10-i1/7379.

[13] S. Rodriguez and C. Mune, "Academic libraries can develop ai chatbots for virtual reference services with minimal technical knowledge and limited resources," vol. 50, no. 3/4, 2022. [Online]. Available: https://doi.org/10.1108/RSR-05-2022-0020.

[14] A. Adike, "Artificial intelligence -analyzing the various parameters considered for ai chatbots," *International Journal of Management IT and Engineering*, vol. 13, pp. 26–31, Dec. 2023.

[15] A. M. B, C. S. Lumingkewas, and A. Rofi'i, "The implementation of user centered design method in developing ui/ux," *JISTE (Journal of Information System, Technology and Engineering)*, vol. 1, no. 2, pp. 26–31, 2023, ISSN: 2987-6117.

[16] E. Çalışkan, "Exam preparation with artificial intelligence: Examination for building material course with chat gpt," *International Conference on Innovative Academic Studies*, vol. 1, Oct. 2023. DOI: 10.59287/icias.1581.

[17] A. Koubaa, "Gpt-4 vs. gpt-3.5: A concise showdown," Apr. 2023. DOI: 10.36227/techrxiv.22312330.v2.

[18] S. Kuraku, F. Samaah, D. Kalla, and N. Smith, "Study and analysis of chat gpt and its impact on different fields of study," Mar. 2023.

[19] H. Liu, R. Ning, Z. Teng, J. Liu, Q. Zhou, and Y. Zhang, "Evaluating the logical reasoning ability of chatgpt and gpt-4," Apr. 2023.

[20] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," 2023. arXiv: 1706.03762. [Online]. Available: https://arxiv.org/abs/1706.03762.

[21] M. M. A. Abdelmoiz, M. M. M. Mostafa, and T. H. A. Soliman, "Developing an educational chatbot for scientific data management course using dialogflow," *Applied Mathematics and Information Sciences*, vol. 18, no. 3, pp. 629–640, 2024.

[22] H. Collier, "Ai: The future of social engineering!" *European Conference on Cyber Warfare and Security*, vol. 23, Jun. 2024. DOI: 10.34190/eccws.23.1.2117.

[23] T. N. Fitria, "Using chatbot-based artificial intelligence (ai) for writing an english essay: The ability of chatgpt, perplexity ai, and chatsonic," *Journal of Language Intelligence and Culture*, vol. 6, no. 2, pp. 103–128, Aug. 2024. DOI: 10.35719/jlic.v6i2.139.

[24] S. Gaikwad, A. Iyer, K. Talluri, and P. Salve, "Sahara: Virtual companion -exploring multimodal empathetic conversational ai leveraging ensemble learning with humor," *International Journal of Science and Research (IJSR)*, vol. 13, May 2024. DOI: 10.21275/SR24419013403.

[25] A. Gajjar, R. Prem Kumar, E. Paliwoda, *et al.*, "Usefulness and accuracy of artificial intelligence chatbot responses to patient questions for neurosurgical procedures," *Neurosurgery*, vol. 95, Feb. 2024. DOI: 10.1227/neu.0000000000002856.

[26] Z. Muhammad Zayyanu Ph.D, "Revolutionising translation technology: A comparative study of variant transformer models -bert, gpt and t5," *Computer Science and Engineering An International Journal*, vol. 14, pp. 15–27, Jun. 2024. DOI: 10.5121/cseij.2024.14302.

[27] I. Nikitina and T. Ishchenko, "Chat gpt in the paradigm of modern education," *Scientific Journal of Polonia University*, vol. 61, pp. 100–106, Mar. 2024. DOI: 10.23856/6112.

[28] V. Noroozi, Z. Chen, S. Majumdar, S. Huang, J. Balam, and B. Ginsburg, "Instruction data generation and unsupervised adaptation for speech language models," 2024. arXiv: 2406.12946. [Online]. Available: https://arxiv.org/abs/2406.12946.

[29] R. Pereira, I. Reis, V. Ulbricht, and N. dos Santos, "Generative artificial intelligence and academic writing: The use of chatgpt," Feb. 2024.

[30] E. Sharkey and P. Treleaven, "Bert vs gpt for financial engineering," 2024. arXiv: 2405.12990 [q-fin.ST]. [Online]. Available: https://arxiv.org/abs/2405.12990.

[31] A. Skuridin and M. Wynn, "Chatbot design and implementation: Towards an operational model for chatbots," *Information*, vol. 15, no. 4, 2024, ISSN: 2078-2489. DOI: 10.3390/info15040226. [Online]. Available: https://www.mdpi.com/2078-2489/15/4/226.

[32] R. E. Turner, "An introduction to transformers," 2024. arXiv: 2304.10557. [Online]. Available: https://arxiv.org/abs/2304.10557.

[33] D. Yang, D. Zhu, H. Gai, and F. Wan, "Semantic similarity caculating based on bert," vol. 20, no. 2, 2024. DOI: 10.52783/jes.1099.

[34] K. Yifanwu, B. Bill, H. Gordon, and C. Mark, "Perplexity: How to calculate perplexity to evaluate the confidence of generated text," May 2024. [Online]. Available: https://docs.kolena.com/metrics/perplexity/.