CZ4052 Cloud Computing Assignment 2

1. Introduction

Software-as-a-Service(SaaS)

Software as a service (SaaS) is a software distribution model in which a cloud provider hosts applications and makes them available to end users over the internet.

Large Language Model (LLM)

A large language model (LLM) is a type of artificial intelligence (AI) program that can recognize and generate text, among other tasks. LLMs are built on machine learning specifically, a type of neural network called a transformer model.

In this project, we will explore on the use of SaaS and LLM functions to create a simple quiz generator chatbot for CZ4052 Cloud Computing module. This project will focus on exploring the parameter tuning of LLM functions and prompt engineering to generate a decent quiz.

We will use **Nemobot** platform which is a SaaS platform that allows users to create chatbot with LLM functions using **OpenAI API**.

2. Explore on Large Language Model Functions

LLM function is a function that is able to recognize and generate text based on the used LLM model. In the LLM function in **Nemobot** platform, we mainly have **5 parameters** shown below:

a. Max Response

Max response parameter defines the maximum number of unit text in the generated response. By setting the max response value, you can limit the length of the generated output. For instance, if the max response value is set to 50, the model will generate a response containing a maximum of 50 unit text.

b. Temperature

Temperature is a parameter that controls the randomness of the generated output where a higher temperature value leads to more randomness. Adjusting the temperature allows you to influence the creativity and exploration of the model.

c. Top P

Top p, also known as nucleus sampling, is another hyperparameter that controls the randomness of language model output. It sets a threshold probability and selects the top tokens whose cumulative probability exceeds the threshold. The model then randomly samples from this set of tokens to generate output.

d. Frequency Penalty

Frequency penalty applies a penalty on the next token proportional to how many times that token already appeared in the response and prompt. This setting reduces the repetition of words in the model's response by giving tokens that appear more a higher penalty.

e. Presence Penalty

Presence penalty also applies a penalty on repeated tokens but, unlike the frequency penalty, the penalty is the same for all repeated tokens. This setting prevents the model from repeating phrases too often in its response.

Experiment on tuning different parameters:

We experimented on manipulating parameters with different values shown in table below.

Parameters	Value 1	Value 2
1. Max Response	50	500
2. Temperature	0.2	0.7
3. Top P	0.2	0.95
4. Frequency Penalty	0	1.5
5. Presence Penalty	0	2.0

The result of the generated text can be found in **Appendix I** to **Appendix V.**

Observation:

1. The higher the Max Response, the longer the generated text
2. The higher the Temperature, the more randomized the generated text

3. The higher the Top P, the more randomized the generated text
4. By having frequency penalty larger than 0, we can prevent generated text from having too many repeated words.
5. By having presence penalty larger than 0, we can get a more diverse text.

In summary, I think having high values of Max Response, Temperature and Top P is good for our quiz generator, frequency penalty and presence penalty might affect the quality of the generated quiz due to the limitation of not having too much repeated words, hence it will be set to 0.

3. Implementation of a simple Quiz Generator Chatbot

In this section, I will briefly talk about how I utilized LLM functions in order to create a simple quiz generator chatbot.

In the chatbot, 4 LLM functions shown as table below are used:

LLM Function	Description
checkAction	To determine what user wanted to ask from chatbot, return 1 if user asked for answer, return 0 if user asked to generate quiz and return -1 otherwise.
generateQuiz	To generate cloud computing quiz based on given guidelines to LLM and available topics
generateAnswer	To generate answer with explanation for the given quiz
chat	To help user answer some general questions.

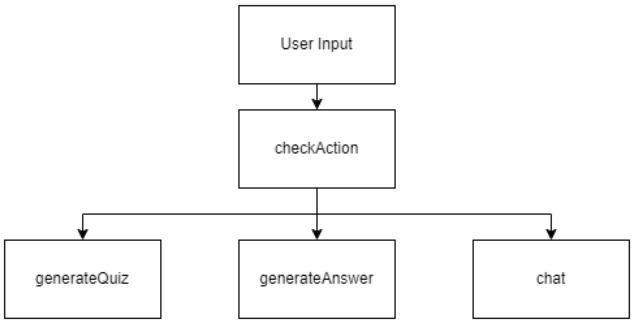

Figure 3.1 Architecture of the quiz generator chatbot

Figure 3.1 shows the simple architecture of the quiz generator chatbot. We will first used checkAction function to determine what user wanted to do and use specific LLM functions for respective task.

And we are able to get a simple quiz generator chatbot that can both generate quiz and answer and help to answer some general questions.

Each LLM functions will perform their task based on the task determined by using the pre-assigned prompt.

4. Explore on Prompt Engineering Large Language Model Functions

Prompt engineering refers to the process of designing and refining prompts or instructions given to artificial intelligence (AI) models to elicit specific desired responses. It's a critical aspect of leveraging AI models effectively for various tasks. A well-crafted prompts can guide AI models to generate highly precise outputs tailored to the user's needs.

In this section, we will focus on explore refining the prompt of generateQuiz and generateAnswer LLM functions with the assistance of ChatGPT to generate quality quiz and answer.

With the help of ChatGPT, I found out that there are several factors that will affect the response of LLM. The main factors are shown as below:

1. Prompt Structure
2. Prompt Length
3. Prompt Clarity and Specificity

Hence, a well structured prompt with clear instructions and moderate length is preferred.

I crafted the prompt by using structure of mainly 2 sections, 1) Task and 2) Guidelines, other sections are added based on the requirements of the task of certain LLM functions. Task is about the task that has been assigned to the current LLM function. Guidelines is about how the response should be generated (format, condition, etc.).

In generateQuiz LLM function, I have experimented and added the following prompt to get a decent generated quiz:

1. I mentioned the available topics from each lecture and set a rule that if user did not specify which lecture they wanted or specify a lecture that is not mentioned here, a quiz that covered random available topics will be generated.
2. For each quiz, there will be only 3 questions, and each question will have 4 answer choices, due to the limitations maximum value of max response is 4096 for a free OpenAI API key, so that a quality quiz can be generated.
3. I introduced the variety of questions types (MCQ, True/False, etc.) by mentioning it in the guidelines.

In generateAnswer LLM function, I experimented and added the following to get a quality response for generated answer:

1. I strictly mentioned that the answer generated should be relevant to the given question in order to prevent unrelated responses from LLM.
2. I mentioned that LLM should give a response with explanation of why the answer is chosen and it should be concise and informative responses.

And in both LLM functions, I mentioned that the format of the response should be in Markdown format so that the response is in a structured format and prevent user from getting different format which might affect the quality of the quiz and answer.

The finalized prompt can be found in **Appendix VI** and examples of generated quiz and answer can be found in **Appendix VII.**

Exported json file which can be load and test on **Nemobot** platform has been uploaded to GitHub repository in **Appendix VIII.** Please feel free to test it out.

5. Conclusion

In conclusion, we have learned about the use of the SaaS platform, **Nemobot** to create a simple and responsive quiz generator chatbot, and by explore tuning parameters of LLM functions and prompt engineering, we can further understand importance of both parameters and prompt in LLM functions. We are able to get the optimal parameters and well-crafted prompt for generating a quiz and answer for CZ4052 Cloud Computing course.

6. Appendix

Appendix I: Comparison of generated text using 50 max response and 500 max response

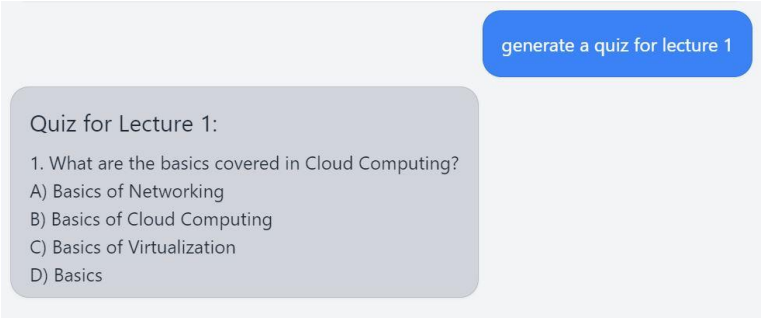

Figure 2.1 Generated Text using 50 max response

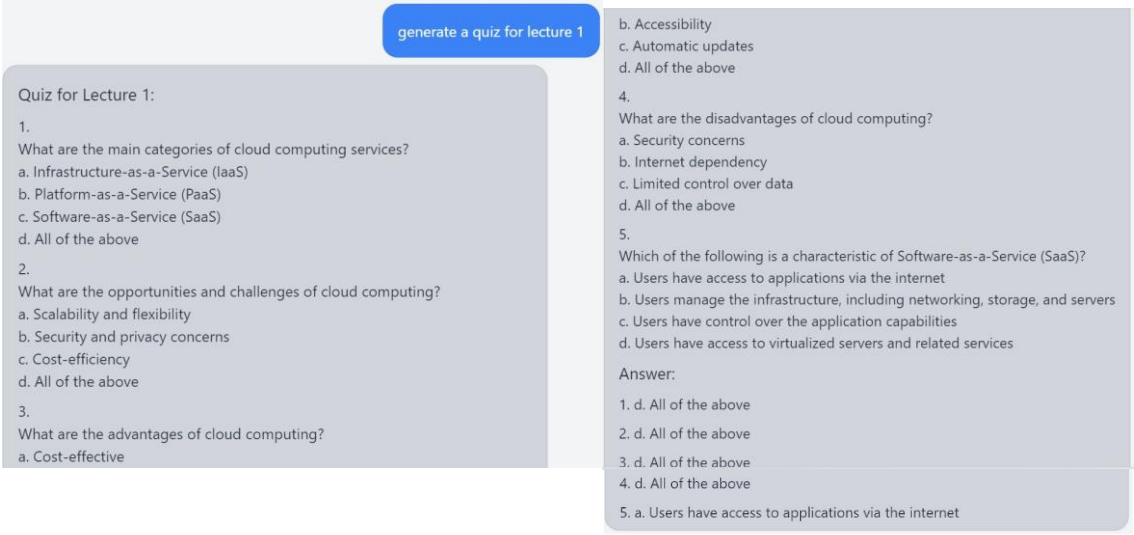

Figure 2.2 Generated Text using 500 max response

Appendix II: Comparison of generated text using 0.2 Temperature and 0.7 Temperature

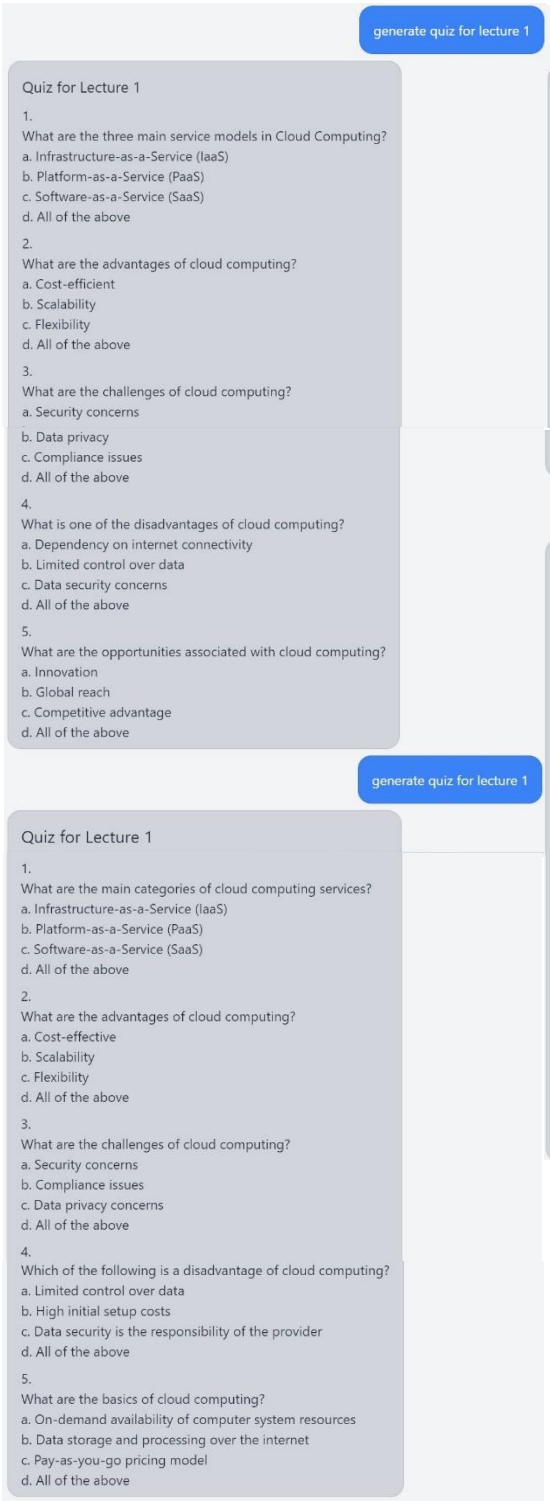

Figure 2.3 Generated text using 0.2 temperature

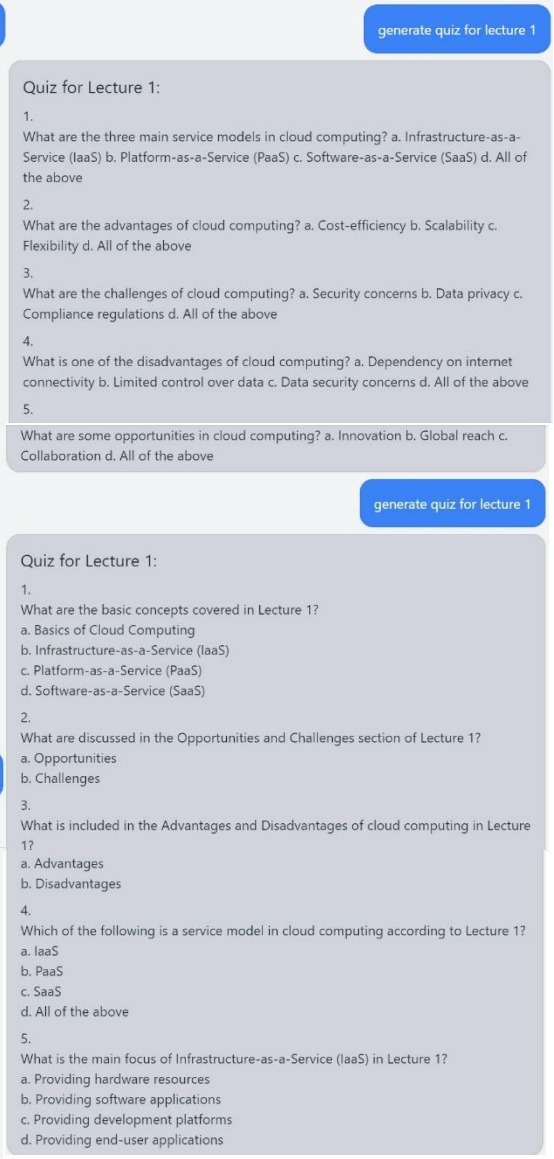

Figure 2.4 Generated text using 0.7 temperature

Appendix III: Comparison of generated text using 0.2 Top P and 0.95 Top P

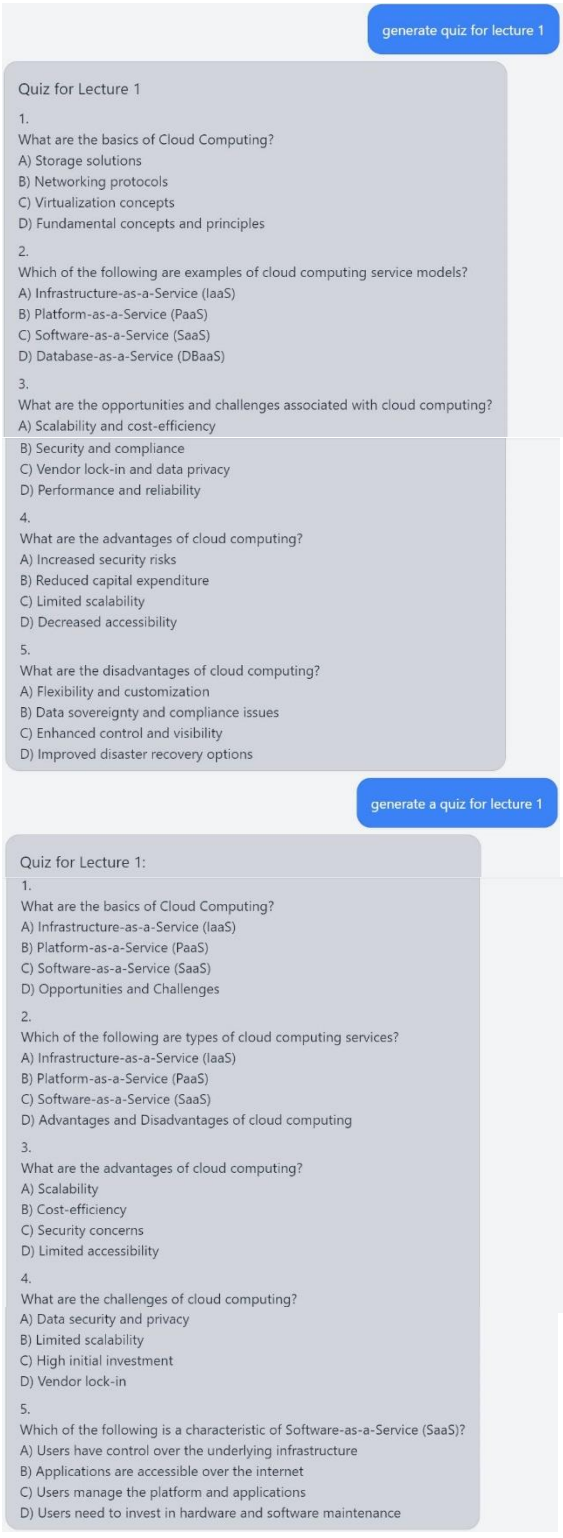

Figure 2.5 Generated text using 0.2 Top P

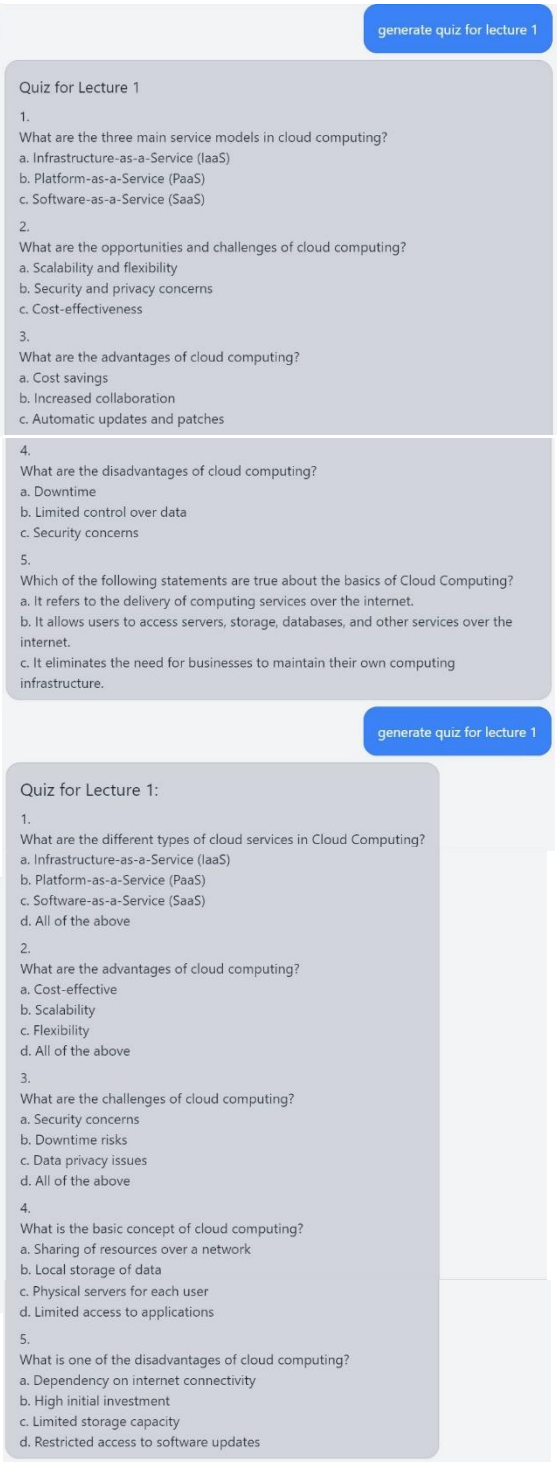

Figure 2.6 Generated text using 0.95 Top P

Appendix IV: Comparison of generated text using 0 Frequency Penalty and 1.5 Frequency Penalty

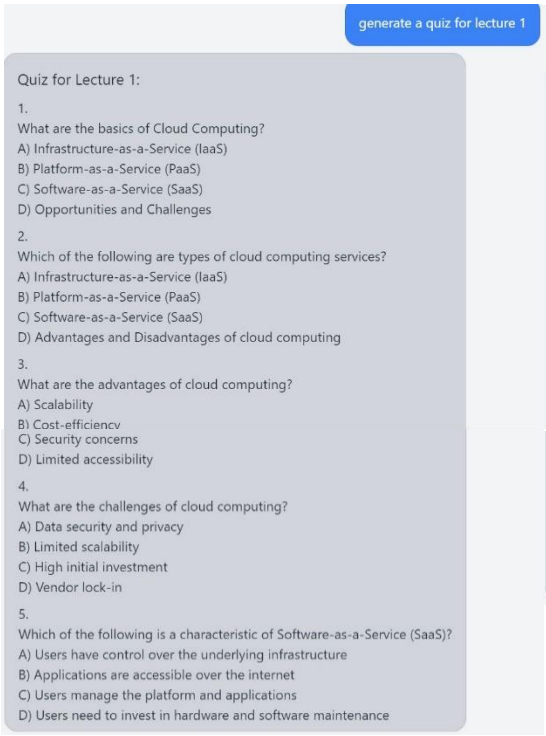

Figure 2.7 Generated text using 0 Frequency Penalty

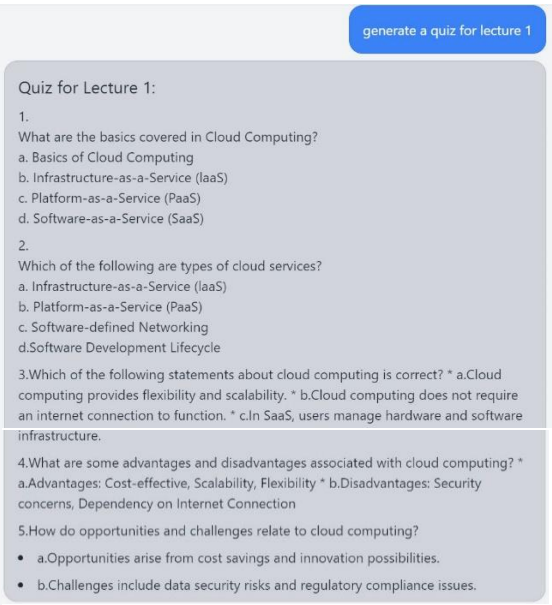

Figure 2.8 Generated text using 1.5 Frequency Penalty

Appendix V: Comparison of generated text using 0 Presence Penalty and 2.0 Presence Penalty

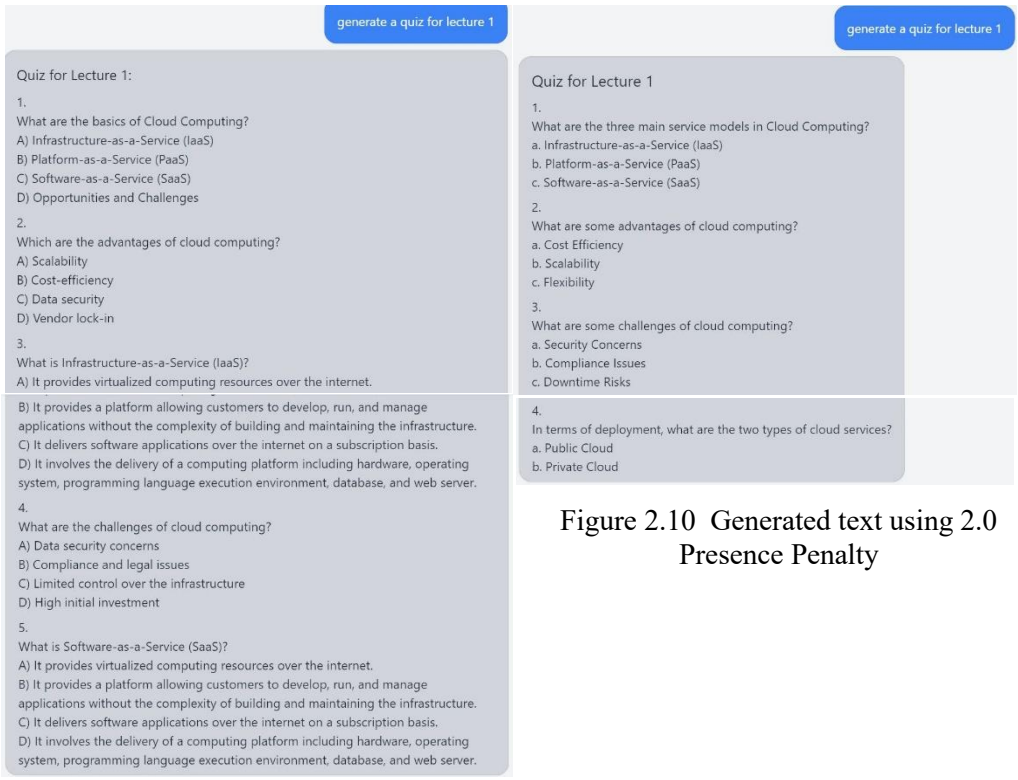

Figure 2.9 Generated text using 0 Presence Penalty

Figure 2.10 Generated text using 2.0 Presence Penalty

Appendix VI: Prompt used by each LLM Functions

You are a helpful assistant but also a quiz generator for CZ4052 Cloud Computing Course.

Your task is to help to answer queries from user.

Your name is Nemobot.

Your reply should be concise, clear, informative and logical.

When you are asked about a question, you MUST think step-by-step. Explain each step in an ordered list.

Your reply should be in Markdown format.

Figure 4.1 chat LLM function Prompt

Your jobs is to determine whether user asked to generate quiz or asked for quiz answers or asked for other question.

Return 1 if user asked for quiz answer.

Return 0 if user asked to generate quiz or what are the available lectures/topics.

Return -1 if otherwise.

Figure 4.2 checkAction LLM function Prompt

Task: Your task is to generate answer based solely on the questions and answer asked by the user. Your answers should be directly relevant to the provided question and should not introduce new topics or information.

Guidelines:

Your reply should be in Markdown format.

You should give explanation on the correct answer

Provide concise and informative responses.

Stick to the topic of each question without diverging into unrelated areas.

Avoid introducing new questions or information not directly requested by the user.

Figure 4.3 generateAnswer LLM function Prompt

Task: Your task is to create a quiz consisting of only questions and answer choices without revealing the correct answer. Each question should be clear, concise, and relevant to the specified topic.

Guidelines:

1. Format: Present each question as a standalone item followed by its answer choices, with each on a separate line. Your reply should be in Markdown format.

2. Clarity: Ensure that each question is understandable without additional context.

3. Relevance: Keep the questions focused on the selected topic.

4. Variety: Include a variety of question types (e.g., multiple choice, true/false, short answer) if possible.

5. Number of Questions: Generate 3 questions for the quiz.

6. Number of Answer Choices: Generate at least 4 answer choices for each question

Available Topics:

1. Lecture 1:

a. Basics of Cloud Computing,

b. Infrastructure-as-a-Service (laaS) ,

c. Platform-as-a-Service (PaaS),

d. Software-as-a-Service (SaaS),

e. Opportunities and Challenges,

f. Advantages and Disadvantages of cloud computing

2. Lecture 2:

a. Basic concepts of network and internet

b. Layering architecture of network and internet

c. TCP/IP

d. TCP Congestion Control Algorithm(Additive Increase Multiplicative Decrease)

3. Lecture 3:

a. Concepts of Virtualization

b. Virtualization architecture

c. CPU and OS basics

d. Types of CPU virtualization

e. Cloud infrastructures

f. RAID (RAID (Redundant Array of Inexpensive Disks)

4. Lecture 4:

a. Why need CPU scheduling?

b. Simple earliest deadline first - schedulability, time granularity, work-conserving

c. Credit scheduler

d. Stride scheduler

5. Lecture 5:

a. Human-assisted Computation: Crowdsourcing (CAPTCHA, reCAPTCHA)

b. Human-assisted Computation: Challenge-Response System

c. Human-assisted Computation: Autograding

6. Lecture 6:

a. CAP Theorem

7. Lecture 7:

a. Page Rank Algorithm

b. PageRank: Random Surfer Model

User Instructions: Please choose one of the predefined topics listed above by specifying the corresponding number when submitting your request. If you do not choose a topic or select a topic not listed here, the AI will generate questions based on a random topic from the predefined options.

Figure 4.4 generateQuiz LLM function Prompt

Appendix VII: Example Quiz and Answer Generated

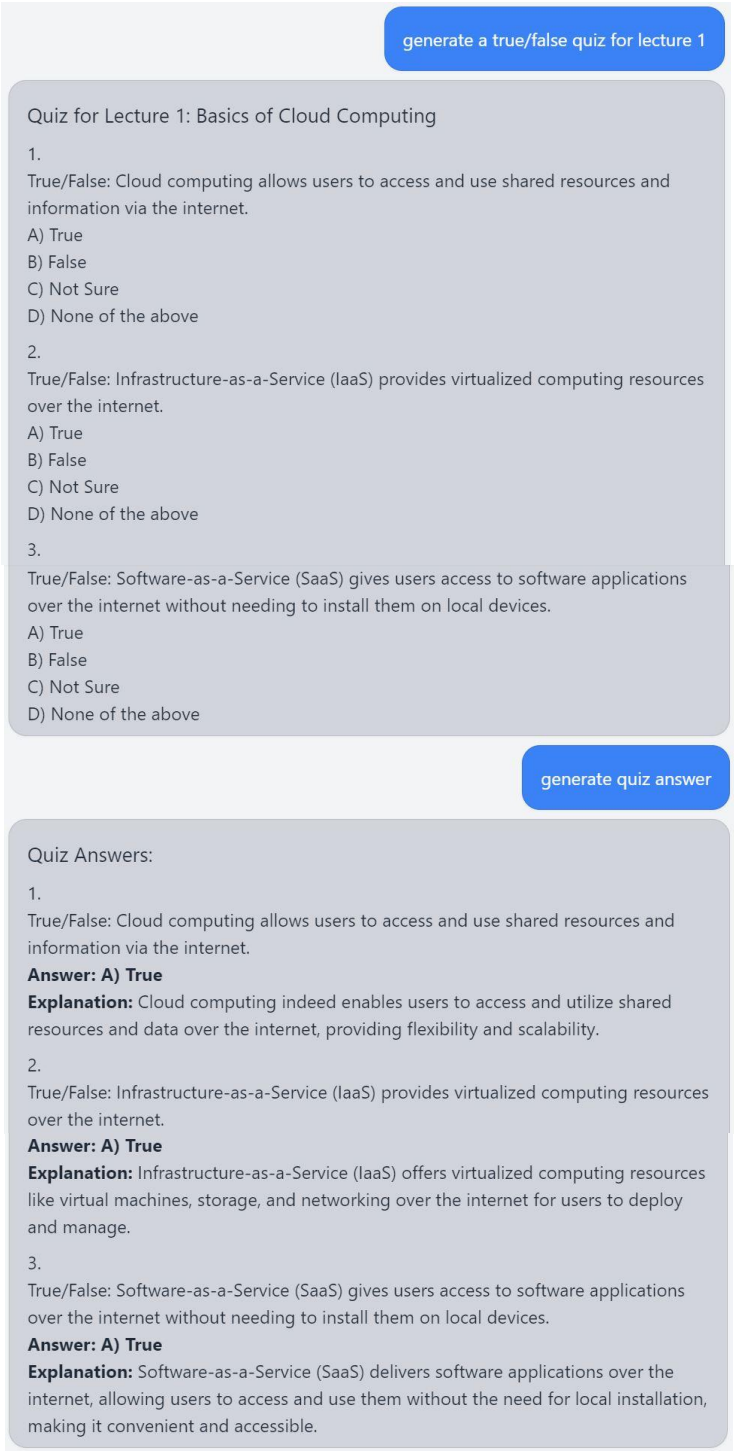

Figure 4.5 Example of Generated Quiz and Answer

Appendix VII: GitHub Repository of Exported Nemobot json file

https://github.com/zijian99/CZ4052_Assignment2_NemobotQuizGen