UNIVERSITY OF CALIFORNIA

SANTA CRUZ

CSE 240 TUTOR BOT WITH A COMPREHENSIVE ANALYSIS

A project submitted in partial satisfaction

of the requirements for the degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE AND ENGINEERING

by

Richard Ho
[rho11@ucsc.edu]

Spring 2024

The Master's Project is approved by:

Dr. Leilani Gilpin, Project Chair

Dr. Razvan Marinescu, Reader

Peter Biehl
Vice Provost and Dean of Graduate Studies

Abstract

In this project, I propose an interactive tutoring system that is specialized for AI courses. I conducted Prolog analysis to compare human answers and artificial intelligence answers for five questions in the CSE 240 graduate course. Prolog was utilized to validate correct answers from ChatGPT. We aggregated a set of ground truth answers, which are correct. We use prolog to validate that the ChatGPT outputs mirror our ground truth answers, so that there are no hallucinations. To do this experiment, I used the OpenAI API version 3.5 to generate samples. I used prolog to validate the samples "truthfulness" and I used word scores, BLEU and ROGUE, to assess the quality of these two AI models respectively. In future work, we will design a user study to assess the quality of my tutor bot contribution. My research opens a new area of interactive tutoring systems which are validated by logical inference rules.

Introduction

Students do not have enough individualized support in core classes. A vast majority of undergraduate classes have about four TAs with an instructor responsible for grading and tutoring large numbers of students, which is about 260 students for instance. Each TA should work forty hours per week from their university. They may work greater than forty hours a week for hosting additional office hours to accommodate stressful students closer to the assignment deadlines in their class. Four TAs spent about 160 hours per week of TA time overall. Every TA spends thirty-five minutes per student in a typical office hour for diagnosing bugs on their program on a weekly basis. Currently, there is a low TA ratio to assist students on their assignments in a large class because students may not be able to receive their feedback thoroughly from their TAs. When students are struggling with their assignment, they often visit the discussion section to obtain assistance from their TAs to perform well in their assignments and exams. However, each TA cannot give every student individualized support. Furthermore, students rely heavily on their TAs on their questions to understand difficult concepts, and TAs had to face a heavy burden on answering students' respective questions. This leads to burnout, and undue burden on teaching assistants to provide constant, consistent student support (Berta and Pembridge, 2019).

One way to reduce the teaching burden is to develop interactive teaching tools. An educational chatbot is useful for answering specific questions on providing appropriate hints guiding them towards answers similar to a TA's office hours session. Furthermore, it will reduce the burden for teaching staff of the overwhelming amount of students' questions to answer in a computer science class. A possible solution is to develop an interface that is needed to validate chatbot answers. It will be useful for answering specific questions on providing appropriate hints guiding them like a TA's office hours (VanLEHN, 2011).

This work is inspired by similar work at Harvard University to develop a teaching chatBot. From "Harvard's New Computer Science Teacher Is a Chatbot", Harvard introduced their own tutor bot on giving hints about their assignments and concepts on the Introduction of Computer Science, which is abbreviated as CS 50 course. They had developed their own custom language model, for CS 50 bot, to assist students on leading an answer to their assignments. A Harvard teaching staff described their AI tutor bot would give students appropriate feedback on their code. Even though the CS 50 chatbot was developed with similarities from ChatGPT, Harvard teaching staff may have used validation techniques to ensure their bot can have functionality to answer respective common questions for their chat bot with generative AI approach. Thus, an generative AI approach will reduce the overwhelming workload on answering questions about their respective questions in their assignments. (Dreibelbis, E. (2023, June 22)). To combat workload at UCSC, we have developed a similar chatBot for advanced AI courses.

During the Winter 2024 quarter, I had conducted research on ChatGPT and OpenAI API version 3.5 respectively for analysis of Artificial intelligence logical reasoning. I also utilized prompting strategies to generate Prolog code to validate their logical reasoning compared to human answers.It allows me to contrast my answers, which are categorized as ground truth answers, and AI answers in my graduate artificial intelligence (CSE 240) class. I had learned about integration of chatbot with python script along with creating a crowdsourcing website. My hypothesis is that both API version 3.5 and ChatGPT version 3.5 have similar logical reasoning for answering common questions for CSE 240. However, Openai's API gave specific and general answers than ChatGPT's vague answers. ChatGPT uses their own database of internet resources for their answers. An investigation of ChatGPT's respective answers to these five questions on analysis of these AI's perspectives contrasted to students' perspectives. Thus, there is a need for a comparison metric developed for further assessment of these respective AI models described later in my research for next quarter.

During this Spring 2024 quarter, I used two metrics: ROGUE and BLEU scores, to verify hypotheses of narrative similarity for my respective LLM models. The first set of experiments was to have the prediction sentence begin at the 2nd trial and reference sentence begin at the first trial respectively for my API trials. The second set of experiments was the reverse; starting from the first set of experiments. I conclude that ROGUE is better than BLEU score because ROUGE score can validate longer sentences. For example, question 1, which was on Appendix B, had a ROUGE score than BLEU score with similar length of prediction and reference sentence. However, there is a lack of metrics for user evaluation nor a user interface on development of tutor bot for actual experiment, which would be discussed in my future work.

In summary, my master's project examines the following research questions:

- RQ1: Is ChatGPT an appropriate interactive tutoring system for courses?
- RQ2: How do we evaluate whether ChatGPT's answers are valid or not?
- RQ3: How do we compare and contrast answers from multiple sources (ground truth, generative AI output, etc.)?

Related work

During my research this Winter 2023 quarter, my goal was to conduct an investigation to understand Prolog generation from LLMs to validate answers for CSE 240. This was inspired by the "Different measurements metrics to evaluate a chatbot system" research article, where these authors discussed their investigation on methods to train and adapt to a chatbot for their respective users because they were evaluating their various metrics on their evaluation in their research. The intention of their research was to access their chatbot on interaction for various users in South Africa. Furthermore, these researchers had evaluated their ALICE chatbot on various metrics: Dialogue efficiency in terms of matching type, Dialogue quality metrics based on response type, and Users' satisfaction assessment based on an open-ended request for feedback (Shawar, B. A., & Atwell, E..2007). The dialogue quality metric “efficiency of 4 sample dialogues in terms of atomic match, first word match, most significant match, and no match” (Shawar, B. A., & Atwell, E..2007) of their evaluation in their interaction with users. Instead, in this work, I use ROUGE and BLEU scores as evaluation metrics because they are meant to work on natural language text.

The second evaluation metric of their chatbot, which is their dialogue quality, evaluates “classify responses according to an independent human evaluation of “reasonableness”: reasonable reply, weird but understandable, or nonsensical reply”(Shawar, B. A., & Atwell, E..2007). My proposed comparison metric would utilize the dialogue quality metrics to analyze my collected data from my AI models this quarter. The dialogue quality metrics used in their research was to assess “efficiency of the adopted learning mechanisms to see if they increase the ability to find answers to general user input”(Shawar, B. A., & Atwell, E..2007). My proposed comparison metric is different from their dialogue quality metric due to assessment between ChatGPT version 3.5, Openai’s API version 3.5, and my answers in a graduate class. Furthermore, it would access the accuracy between ChatGPT and API version 3.5 respectively.

User evaluation and creation of user interface are useful for understanding students’ experiences for a tutor bot. According to “RetLLM-E: Retrieval-Prompt Strategy for Question-Answering on Student Discussion Forums,” these researchers discussed their metric for their QA bot for student discussion forums along with their evaluation on their students’ usefulness in their research. They had used ROGUE and BERTscore for their “ground truth” (Norouzi and etc., 2024) on their QA bot on student discussion forums prior to discussion of human evaluation metrics in their research paper. A human evaluation metric had been discussed about “Relevance, Read-ability, Tone, and Factuality of the responses” from a scale of 1-3(low to high evaluation) for students’ feedback on their tutor. Thus, my proposal of utilizing a similar human evaluation metric for my tutor bot on my continuation of my research project beyond my completion of my Masters’ Capstone project.

Methods

In order to collect sample data from ChatGPT and 2 api versions(gpt-3.4 and gpt-3.5), there were 5 questions, which is on Table 1 below, from the CSE 240(Artificial Intelligence Graduate class) with my answers and respective LLM models with respective abridged answers from ChatGPT version 3.5, OpenAI verison gpt-3.4 and gpt-3.5:

- 1)When is DFS better than BFS search?
- 2) What is the difference between a state space graph and a search tree?

- 3) How do Stochastic Hill Climbing and Random Restart Hill-Climbing vary on the hill-climbing augmentation strategies?
- 4) Why do we prefer to change nearly tree-structured CSPs to tree-structured CSPs?
- 5) Suppose h1 and h2 are admissible heuristics. Suppose we are using A*-search. Why is it a good idea to use max(h1,h2) rather than either h1 or h2 alone as the heuristic evaluation function?(Table 1 and Table 2).

Furthermore, these five questions are open-ended questions to examine Natural Language Learning's logical reasoning in contrast to students' common answers. After comparison of human answers and the Language Learning models, an utilization of the ChatGPT and Openai api versions to generate a prolog on having an AI logic reasoning to answer their respective questions. Prolog allows me to validate the artificial intelligence's logical reasoning for CSE 240 questions on both ChatGPT and API version 3.5 in my research. Furthermore, it allowed me to compare their respective prolog rules and my human answers on these questions to validate their answers. Even though the Prolog examined the ChatGPT and Openai's API version 3.5 respectively, there were some differences in the logical reasoning of answering these five questions respectively. This will be discussed in the Results section with experimental results from Appendix C.

In order to analyze the logic of the AI answers for my sample data of CSE 240 questions, closed-ended questions typically have common answers from the internet due to the collected data from ChatGPT and OpenAI. Open-ended questions from CSE 240 would challenge the logic reasoning for the ChatGPT and OpenAI models. The five questions developed throughout the winter quarter for analysis of the logic reasoning and comparing though the AI models:

CSE 240 Prompts and Ground Truth Answers (Table 2)

1)When is DFS better than BFS search?	Never because it depends on various situations to use a respective task.
2) What is the difference between a state space graph and a search tree?	The tree is typically searched depth-first and the nodes are implicit meaning they are generated as needed. The state-space of a dynamical system is the set of all possible states of the system.
3) How do Stochastic Hill Climbing and Random Restart Hill-Climbing vary on the hill-climbing augmentation strategies?	Stochastic hill climbing has a randomized uphill move and varies on the steepness and the random restart hill climbing has a series of hill climbing searches from randomly generated states until a goal is found.
4) Why do we prefer to change nearly tree-structured CSPs to tree-structured CSPs?	Cutset conditioning is to pick a cutset in a tree structure and we can instantiate the cutset in all

	possible ways. Then we compute and solve residual CSP tree structured for each assignment.
5) Suppose h1 and h2 are admissible heuristics. Suppose we are using A*-search. Why is it a good idea to use max(h1,h2) rather than either h1 or h2 alone as the heuristic evaluation function?	A heuristic admissible if $h(n) < h^*(n)$ for A* search for max(h1, h2). The efficiency of the A* search is least time efficient in contrast to other search techniques. The max of h1, h2 is closer to the true cost.

I developed five answers as human answers because they are core questions from CSE 240 in Winter 2023. My intention of classifying my inclass question answers as human answers because most students would learn the material and pay attention to lectures, which are the most logical answers. Furthermore, human answers are classified as ground truth for students' appropriate answers on attending lectures. It had also been utilized for comparing the AI respective answers and logic.

Throughout my research on examining the Prolog differences between ChatGPT and OpenAI’s API versions, there were similar concluding sentences throughout their explanations, such as their phase “in summary”, on questions 2-5. For instance, ChatGPT always concludes with the “In summary, …” for their answers on questions 2-5 and “It's important…” on question 1. In contrast to the ChatGPT answers, OpenAI have abstract answers for respective answers to their questionsFurthermore, a development of a metric on examining these similar word patterns for evaluation on all collected LLM models for next quarter. Despite the minor differences between the human and artificial intelligence logical reasoning from my sample data this quarter, more investigation and analysis needs to be analyzed on the two AI models.

During my research on BLEU scores and ROUGE scores on verifying my hypothesis about my Openai’s API model, BLEU scores discussed their metric evaluation between reference and prediction sentences respectively based on their precisions with a sequence of words, which is n-grams. This metric typically heavily relies on n-grams for reference and prediction sentences for their calculation of narrative similarity. The drawback of using BLEU scores was an inaccurate calculation of narrative similarity for varying amounts of n-grams for either reference or prediction sentences due to heavy dependence of length of sentences for both references and prediction respectively. In contrast to BLEU scores, ROUGE scores discussed a metric for text summarization tasks on their objective of summarizing longer sentences. Furthermore, it measured the similarity between prediction and reference sentences from a word sequence. After evaluating BLEU and ROGUE scores, there was a lack of time to evaluate a user evaluation metric along with an user interface to examine my tutor bot further beyond this Spring 2024 quarter.

Results

The experimental setup for analysis of a LLM model answers respectively for comparison of ground truth answers by asking to generate five versions of Prolog code. A prompting strategy to ask their LLM model to ask multiple questions, such as “What is the Prolog for this chat?”, to ChatGPT on generating Prolog codes with bugs initially. After my initial prompting strategy caused ChatGPT to generate Prolog codes, which had compile errors, the new question was to ask ChatGPT about “Can you translate this answer into Prolog?” to generate Prolog codes to compile properly. The generated Prolog codes and rules for these five questions results are from Appendix C. After my experiment of conducting analysis of both ChatGPT and OpenAi’s API version 3.5 respectively, question two, which discussed from Table 2 of Ground Truth answers, had one rule from OpenAI’s API version 3.5 and seven rules from ChatGPT version 3.5. A lower amount of Prolog rules from these Prolog rules represented a comprehensive understanding of students of their question.

I used ChatGPT to generate a prolog code to analyze the logical reasoning of respective answers on these five questions. ChatGPT generates various outputs of prolog code because they have the ground truth of their answers. Question 4 and question 5 are open-ended questions to assess the ChatGPT’s quality of their answers. Some of the prolog generated would not load, however, asking ChatGPT to generate prolog code after answering the question has valuable results. An initial prompting strategy to generate accurate prolog code was to ask ChatGPT on “what is the prolog for this conversation?”,”What is the prolog for this chat?”, “what is the prolog for this conversation?”, “what is the prolog for when is dfs better than bfs search?”, and “generate prolog” had mixed prolog results. When I utilized the “generate prolog” in a second prompt for my previously generated answers from ChatGPT version 5.0, ChatGPT generated complex prolog codes, which led to major logical errors, for my initial prompting strategy on accessing the differences between artificial intelligence and human’s logic. Furthermore, the debugging process using ChatGPT had initial difficulties with copypasta the prolog error message on attempting to make the code run. The new prompting strategy was to ask ChatGPT to generate clean code by “Can you translate this response into prolog?” allowed me to view multiple rules from Prolog to understand ChatGPT’s AI reasoning. The AI may understand a direct meaning behind the user’s interpretation of the user’s question to address these difficulties from the ChatGPT version 3.5.

The prompting strategy allowed me to evaluate the ground truth between the ChatGPT version 3.5 and Openai’s API version 3.5. It allowed a generation of the Prolog codes to analyze respective rules on respective AI models. Prolog rules represent the process of a LLM model on respective steps for their answers respectively. Less rules represented a clear logic of AI-generated answers. For instance, there were two rules for the ChatGPT’s answer and 4 rules for the OpenAI’s API version 3.5 on question one. Another example was one rule for API and seven rules for ChatGPT on question two for Prolog analysis of ground truth answers. Overall, ChatGPT had difficulty interpreting their question in contrast to their API. The prolog rules for both ChatGPT version 3.5 and Openai’s API version 3.5 are in Appendix C for reference. Both of these samples have the same amount of rules for question four.

My improvement of having the prolog codes worked dramatically on getting reliable results for ChatGPT question 5 answers. When I asked ChatGPT to fix their Prolog code on compiling, there were similar generations of Prolog code with errors. When I decided to revise my prompting strategy, my new prompting strategy was to ask ChatGPT “Can you turn this response into a prolog?” until I get a satisfactory answer, which is a generated prolog code that runs properly with no errors. In this situation, ChatGPT generated Prolog code for their answers for analysis of their answers. Even though there were minor bugs from ChatGPT’s generated Prolog code of questions 3-5, I asked ChatGPT to provide a sample answer of a function in prolog to examine the compile run. After asking ChatGPT to fix “Please rewrite this code to resolve this error message: ERROR: Unknown procedure: heuristic_evaluation/2 (DWIM could not correct goal)” on question 5, I finally got runnable Prolog code after utilization of my revised prompting strategy.

The ChatGPT version 3.5 gives specific information on answering open-ended questions(question 1 and question 2) unlike their API gpt-3.5-turbo-instruct. Openai’s API version 3.5 in the text completions in the OpenAI has an abstract version for answering open-ended questions. As demonstrated in the prolog cross references with our respective AI tools, there are less rules in the API in contrast to the ChatGPT. Openai’s API version 3.5 contained less rules created for answering open-ended questions unlike ChatGPT. For instance, question 2, which is from Appendix A Table 1 and Table 2, have one rule from the openai’s API version 3.5 turbo and seven rules from the ChatGPT’s version 3.5. ChatGPT utilized their database to explain their vague answers in contrast to their API version 3.5 turbo version’s summarized approach of their logical reasoning. Both ChatGPT version 3.5 and OpenAI API version 3.5 have similar logical reasoning on answering question 4 because the assessment of both AI LLN models has four rules. Thus, ChatGPT and OpenAI API version 3.5 respectively can have similar reasoning to approach the questions by validating the prolog rules in Appendix C.

After conducting ten trials on respective questions for the API to examine the reoccurring results from the previous run, there are some results that have long answers and short abstract answers thereafter. Original version of the API is gpt-3.4-turbo-preview prior to the release of gpt-3.5-turbo-instruct model. Furthermore, I decided to compare the gpt-4-turbo-preview and gpt-3.5-turbo-instruct versions respectively for the API from the OpenAI website. From the analysis between the gpt-3.4 API and gpt-3.5 API, the gpt-3.4 API may act more like the ChatGPT version because they give specific information about the concepts for open-ended questions, such as question 1. Due to lack of ample time to conduct more trials for the gpt-3.4 API, I conducted 2 trials for each question to examine the quality of the answers for comparison of the gpt-3.5 and ChatGPT version 3.5. Thus, open-ended questions sometimes produced consistent results and ⅘ questions with abstract answers along with heavy context answers.

During this Spring 2024 Quarter, the first set of BLEU experiment was to have prediction sentences as my second trial and reference sentence as my first trial, which had been reutilized from last quarter, initial pair. The round robin is an alternation of initial prediction and references sentences from an initial set of experiments from Appendix B. As there are about nine respective pairs for calculation of BLEU scores to measure narrative similarity, there are various AI generated sentences in my dataset that represent prediction sentences and reference sentences. A second set of experiments would have a round robin from the first set of nine pairs respectively for calculating the BLEU scores. When I calculated ROUGE scores to evaluate narrative similarity of the API's sentences, I used the same procedure from calculation of BLEU scores. The calculations of BLEU scores and ROUGE scores respectively for nine pairs of sentences for two sets of trial runs were shown in Appendix B. The BLEU score was 0.61 and ROGUE score was 0.81 for question one showing that ROGUE is better than BLEU score.

Discussion

During the Winter 2024 quarter, I had conducted analysis of ChatGPT and Openai API version gpt-3.5-turbo instruct & gpt-3.4-turbo instruct to access their various answers with my respective human answers of my five questions from CSE 240. My initial prompting strategy had led to obstacles due to the generation of Prolog codes that led to initial coding errors because I had used various questions on prompting unexpected generated Prolog codes with bugs. After I fixed my prompting strategy to ensure ChatGPT generates appropriate Prolog code which can be loaded,I used the Prolog rules and facts compared with my human answers, which was developed with my knowledge of CSE 240 class. When I conducted ten trials on the gpt-3.5-turbo, there were few recurring word choices that had occured in my trial runs in all of my five CSE 240 questions : “In summary”, “When using”, and “Overall”. Thus, both ChatGPT and Openai's version gpt-5-turbo have similar logic for answering my five CSE 240 questions.

My limitation of my research for the Winter 2024 quarter was no comparison metric to measure the accuracy of answering questions appropriately. Furthermore, conducting ten trials to have similar results from previous trials is redundant on accessing the accuracy of the OpenAI gpt-5-turbo. Despite my success in developing a prompting strategy for assessment of human answers, there were consistent word choices, such as “In summary”, reoccured on some of my trial runs. The prompting strategy allowed me to compare the logical reasoning between artificial intelligence answers and human answers. However, it doesn't allow any comparison metric to assess the accuracy of the natural language models respectively. Thus, a development of a comparison metric along with other evaluation metrics assist the assessment of how these respective AI models answered CSE 240 questions.

The ChatGPT's logical reasoning from Prolog for answering these five questions from CSE 240 throughout this quarter led to interesting findings. Furthermore, I had conducted additional research on OpenAI's API versions to compare their answers along with their logic with Prolog similar to my ChatGPT's procedure. After an analysis of my sample collected data this quarter, ChatGPT has to utilize internet resources on forming their answers in contrast to students' knowledge from lectures. During the Spring 2024 quarter, a comparison metric would be conducted for investigation on analysis of respective AI models from collected data. Furthermore, there is a need for more word metrics based on appropriate evaluations of the AI models, such as blue choice.

During this Spring 2024 quarter, I conducted an analysis of narrative similarities of the Openai's API between ROUGE and BLEU scores. BLEU scores performed efficiently on a situation of similar length of sentences(n-grams) for each pair, which represent prediction and reference sentences. For instance, the third pair of sentences from question three, which contained the fourth trial as a prediction sentence and the third sentence as the reference sentence, had about 0.60 BLEU score. The length of prediction sentences and reference sentences had been calculated using words as their n-gram respectively. In this example, the prediction sentence had about 91 words and the reference sentence had precisely 71 words on their third pair for CSE 240 human question three. The result of common BLEU score is 0.6 indicating there is a strong possibility that there is a narrative similarity for AI generated sentences. However, a weakness of BLEU scores heavily rely on common phases of two AI-generated sentences along with similar length of sentences caused to have lower scores than anticipated. For instance, the first pair of sentences, prediction sentence was second sentence and reference sentence was first sentence, from question three had a BLEU score of 0.14. The length of the prediction sentence was 461 words and the reference sentence length was 141 words respectively.

When there was a calculation of results of using ROUGE scores with similar experimental procedures from BLEU scores, ROUGE scores had better calculations than BLEU scores due to interpretation of longer sequences of sentences. It had utilized ROUGE L scores for the intention of examining the narrative similarity more efficiently than BLEU scores. ROUGE L scores measured the common subsequences to evaluate the sentence comparison sequence between prediction and reference sentences unlike BLEU score metric evaluation. Furthermore, it would be expressed for my On a similar experiment from the third pair of sentences from question three, the ROUGE L scored about 0.85 contrast to BLEU scores of 0.60. Another instance of ROUGE scores had higher narrative similarity than BLEU scores was the first pair of sentences from question three was 0.68 contrast to BLEU scores of 0.14. Thus, ROUGE scores have a stronger evaluation metric than BLEU scores for analysis of narrative similarity of my CSE 240 human questions of a tutor bot.

Future Work

There are multiple other areas to build upon in this research. One such idea is to design, implement, and evaluate a user study for validating that my system's responses are helpful to students. I propose to develop a user study in order to have a quantitative human evaluation of Factuality, Relevance, and Readability(Ranade, G., Norouzi, N., and al., 2024).   Factuality would discuss AI generated sentences and give meaningful hints to students for answering their question. An example of factuality is for an AI generated answer on appropriate hints and not giving actual answers similar to TA/ professor's office hours. Relevance evaluates on a tutor bot can answer the student's question correctly and on topic. For instance, an AI tutor bot needs to give course-related answers and avoid unrelated course topics, such as Operating systems class. Readability evaluation would assess a user's comprehension of my tutor bot's advice on their homework question. For instance, students should understand the AI generated answers, which is related to their lecture and assignment, in a Computer Science class.

The user study would be conducted in a classroom setting with about thirty through forty participants. Participants would be recruited from a university's computer science class with consent to participate in an user study. Users would be presented with five questions, such as “What is the difference between a state space graph and a search tree?” from Appendix A and Table 2: CSE 240 Prompts and Ground Truth Answers, and they would be instructed to use our tool for help. We would record the number of times they used our tool, the number of queries, and have a post experiment survey to evaluate the user’s experiences based on our tool with quantitative human evaluation for my user study. The post experiment survey would have questions from a quantitative human evaluation with factuality, relevance, and readability as my rubric in my user study.

The expected outcome of my user study is to evaluate the interactions between our tutor bot and students in the university setting. Furthermore, we wanted to examine the user's comprehension of AI generated sentences from the user's course-related questions. Our goal with the user study is to answer the following research questions:

RQ1: Are LLM outputs useful for answering key AI questions?

RQ2: Do LLM outputs help students answer their questions faster?

RQ3: Do LLM outputs help students achieve learning goals?

These research questions addressed my proposed user study’s evaluations on my ongoing research in public education.

References

1)Dreibelbis, E. (2023, June 22). *Harvard's new computer science teacher is a chatbot*. PCMAG. https://www.pcmag.com/news/harvards-new-computer-science-teacher-is-a-chatbot

2)Different measurements metrics to evaluate a chatbot system
Shawar, B. A., & Atwell, E. (2007, April). Different measurement metrics to evaluate a chatbot system. In *Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies* (pp. 89-96).

3)Mitra, C., Miroyan, M., Jain, R., Kumud, V., Ranade, G. and Norouzi, N. 2024. RetLLM-E: Retrieval-Prompt Strategy for Question-Answering on Student Discussion Forums. *Proceedings of the AAAI Conference on Artificial Intelligence*. 38, 21 (Mar. 2024), 23215-23223. DOI:https://doi.org/10.1609/aaai.v38i21.30368.

4)Berta, M. R., & Pembridge, J. J. (2019, June), *Exploring Burnout among Graduate Teaching Assistants* Paper presented at 2019 ASEE Annual Conference & Exposition , Tampa, Florida. 10.18260/1-2--32802

5)VanLEHN, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, *46*(4), 197–221. https://doi.org/10.1080/00461520.2011.611369

Appendix A

CSE 240 Questions and ChatGPT Version 3.5 Answers (Table 1)

Questions	ChatGPT Version 3.5(Free version)
1)When is DFS better than BFS search?	Depth-First Search (DFS) and Breadth-First Search (BFS) are two fundamental algorithms used in graph traversal. The choice between DFS and BFS depends on the specific characteristics of the problem and the requirements of the application. Here are some situations where DFS might be preferred over BFS: Memory Efficiency:DFS is generally more memory-efficient than BFS. In DFS, you only need to keep track of the nodes along the current path from the root to the current node, while in BFS, you need to store all nodes at the current depth in a queue. If memory is a critical factor, and the depth of the tree/graph is large, DFS may be a better choice.Pathfinding to a Solution:If the goal is to find any solution or path from the start node to the goal node, DFS might be more suitable. DFS tends to go deep into the search space quickly, which can lead to finding a solution faster if the goal is located deep in the tree/graph.Space Complexity:In some cases, DFS has a lower space complexity than BFS, especially when the tree/graph is sparse. This is because DFS doesn't require maintaining a separate data structure to store all nodes at the current depth. Topological Sorting:DFS is often used for topological sorting of directed acyclic graphs (DAGs). It can efficiently find a topological ordering of the nodes, which is useful in scheduling and task dependency problems.It's important to note that the choice between DFS and BFS depends on the specific requirements of the problem at hand. BFS is generally better for finding the shortest path in an unweighted graph and guarantees the shortest path in terms of the number of edges. In contrast, DFS may find a path faster but doesn't guarantee optimality in terms of the number of edges.
2) What is the difference between a state space graph and a search tree?	A state space graph and a search tree are two related concepts used in the context of search algorithms, such as depth-first search, breadth-first search, and A* search. Here are the key differences between them: Representation: State Space Graph:It represents the entire space of possible states and transitions between states in a problem. Nodes in the graph represent individual states, and edges represent possible transitions or actions between states. Search Tree:It is a tree structure generated during the execution of a search algorithm.Nodes in the tree represent the states encountered during the search, and edges represent the actions taken to move from one state to another. Completeness:State Space Graph: It is a complete representation of all possible states and transitions in the problem domain.The state space graph includes every state that can be reached from the initial state by applying valid actions. Search Tree:It is a partial representation of the state space, created dynamically during the search process.The search tree includes only the states that have been explored during the search up to a certain point.

	Exploration vs. Representation: State Space Graph: It provides a static, comprehensive view of the problem space, allowing a global understanding of the problem. It is useful for analyzing the structure of the problem and understanding the relationships between states. Search Tree: It represents the exploration process of the search algorithm. It is dynamic and reflects the states that have been visited and the actions taken during the search. Memory Requirements: State Space Graph: It may require a large amount of memory to represent the entire space of states, especially for problems with a large state space. Search Tree: It only requires memory to store the states that have been explored during the search, making it more memory-efficient in practice. Visualization: State Space Graph: It is a visual representation of the entire problem space, which may be too large to visualize in its entirety. Search Tree: It is a compact representation that visualizes the exploration process, making it easier to understand the search progress. In summary, a state space graph is a comprehensive representation of all possible states and transitions in a problem, while a search tree is a dynamically generated structure that represents the exploration process of a search algorithm within that state space. The search tree is more focused on the states encountered during the search, making it a practical tool for visualizing and understanding the search process.
3) How do Stochastic Hill Climbing and Random Restart Hill-Climbing vary on the hill-climbing augmentation strategies?	Stochastic Hill Climbing and Random Restart Hill-Climbing are two variations of the hill-climbing algorithm with different augmentation strategies. Let's explore how they differ in terms of their hill-climbing augmentation strategies: Stochastic Hill Climbing: In Stochastic Hill Climbing, randomness is introduced during the selection of the next move. Instead of always choosing the best neighboring state, Stochastic Hill Climbing probabilistically selects a neighboring state based on certain probabilities. This randomness allows Stochastic Hill Climbing to explore a broader space, potentially avoiding local optima that a deterministic approach might get stuck in. The degree of randomness can be controlled by parameters like temperature in simulated annealing, where higher temperatures allow more exploration. Random Restart Hill-Climbing: Random Restart Hill-Climbing addresses the issue of getting stuck in local optima by periodically restarting the search from a randomly chosen initial state.

	The algorithm performs multiple hill-climbing searches from different starting points (random restarts). After reaching a local optimum or exhausting a certain number of steps, the algorithm randomly selects a new initial state and starts the hill-climbing process again. This approach aims to explore different regions of the search space and increase the chances of finding a global optimum. Comparison: Exploration vs. Exploitation: Stochastic Hill Climbing emphasizes exploration by introducing randomness during the move selection, allowing it to escape local optima more easily. Random Restart Hill-Climbing focuses on exploration by restarting the search from different initial states, diversifying the search. Memory of Past States: Stochastic Hill Climbing relies on probabilities and may consider the history of states visited during the search through mechanisms like simulated annealing. Random Restart Hill-Climbing does not explicitly maintain memory of past states but achieves diversity through random restarts. Convergence: Stochastic Hill Climbing may converge to a solution more gradually, as the randomness allows it to explore and exploit simultaneously. Random Restart Hill-Climbing is designed to prevent premature convergence by periodically restarting the search, promoting more extensive exploration. Global Optimum: Stochastic Hill Climbing may or may not find the global optimum depending on the degree of randomness and exploration. Random Restart Hill-Climbing increases the likelihood of finding the global optimum by exploring different regions of the search space. In summary, Stochastic Hill Climbing introduces randomness in the move selection process to explore a broader space, while Random Restart Hill-Climbing addresses local optima by periodically restarting the search from different initial states. Both strategies aim to enhance the effectiveness of the basic hill-climbing algorithm in finding optimal solutions.
4) Why do we prefer to change nearly tree-structured CSPs to tree-structured CSPs?	In Constraint Satisfaction Problems (CSPs), a tree-structured CSP refers to a CSP where the constraint graph (graph representing relationships between variables) is a tree. A non-tree-structured CSP has a more complex constraint graph with cycles or more intricate relationships among variables. Here are some reasons why it is often beneficial to transform non-tree-structured CSPs into tree-structured CSPs: Simplicity and Modularity: Tree-structured CSPs are simpler and more modular in nature. The absence of cycles in the constraint graph simplifies the representation of relationships among variables. Decomposing a non-tree-structured CSP into a set of tree-structured subproblems can make it easier to analyze and solve each subproblem independently.

	Efficient Algorithms: Many efficient algorithms for solving CSPs, such as backtracking search and dynamic programming, are designed for tree-structured CSPs. Transforming a non-tree-structured CSP into a tree-structured form allows the application of these efficient algorithms, leading to improved computational efficiency. Consistency and Inference: Tree-structured CSPs often exhibit a property called arc-consistency, which simplifies the constraint satisfaction process. The absence of cycles in tree-structured CSPs makes it easier to perform local consistency checks and infer information about the variables. Parallelization: Tree-structured CSPs naturally lend themselves to parallelization. Subproblems represented by the branches of the tree can be solved independently, allowing for parallel processing and speeding up the overall solution process. Problem Decomposition: Transforming a non-tree-structured CSP into a tree-structured form allows for effective problem decomposition. Decomposing the problem into smaller, more manageable subproblems simplifies the overall problem-solving process. Improved Understanding: Tree-structured CSPs provide a more intuitive and understandable representation of the relationships between variables. The tree structure often corresponds to a hierarchical or sequential arrangement of decisions, making it easier for humans to comprehend. In summary, transforming non-tree-structured CSPs into tree-structured CSPs offers advantages in terms of algorithmic efficiency, problem decomposition, parallelization, and overall simplicity. This transformation can make it easier to apply existing CSP-solving techniques and algorithms, leading to more effective and scalable solutions.
5) Suppose h1 and h2 are admissible heuristics. Suppose we are using A*-search. Why is it a good idea to use max(h1,h2) rather than either h1 or h2 alone as the heuristic evaluation function?	In the context of A* search, where admissible heuristics are used to guide the search algorithm, combining multiple admissible heuristics using the maximum function (max(h1, h2)) is known as taking the "maximum heuristic." The key reason for using the maximum heuristic is related to the concept of dominance and the admissibility property. Here's why it's a good idea to use max(h1, h2) instead of either h1 or h2 alone:Admissibility:Both h1 and h2 are assumed to be admissible heuristics, meaning that they never overestimate the true cost to reach the goal. Using max(h1, h2) ensures that the combined heuristic remains admissible.If h1 and h2 are admissible, then max(h1, h2) is also admissible because it is guaranteed to be at least as good as the better of h1 and h2. Dominance:One heuristic might provide more accurate estimates in certain situations, while the other might be more accurate in different situations. Taking the maximum heuristic helps in capturing the strengths of both heuristics.If h1 dominates h2 in certain states (h1 is always greater than or equal to h2), then max(h1, h2) is essentially equivalent to using h1 in those states.Similarly, if h2 dominates h1 in certain states, max(h1, h2) is equivalent to using h2 in those states.

	Balance:Using max(h1, h2) strikes a balance between the two heuristics. It ensures that the heuristic function is not overly optimistic (underestimating the cost), as it considers the most optimistic estimate among the two. Improved Guidance:The maximum heuristic tends to guide the search more effectively by incorporating the strengths of both heuristics. This can lead to a more informed exploration of the search space. In summary, using max(h1, h2) as the heuristic evaluation function in A* search allows for the combination of multiple admissible heuristics in a way that maintains admissibility and captures the advantages of each heuristic in different parts of the search space. This can lead to improved efficiency and effectiveness in finding optimal solutions.

Appendix B: BLEU and Rouge Scores

Calculating BLEU Score for 10 trials of the API

First question:

Prediction sentence trial 2

Reference sentence trial 1

{'bleu': 0.6144699681759951, 'precisions': [0.8833333333333333, 0.7068965517241379, 0.6607142857142857, 0.6296296296296297], 'brevity_penalty': 0.8607079764250577, 'length_ratio': 0.8695652173913043, 'translation_length': 60, 'reference_length': 69}

Prediction sentence trial 3

Reference sentence trial 2

{'bleu': 0.20098007190633455, 'precisions': [0.34782608695652173, 0.19402985074626866, 0.16923076923076924, 0.14285714285714285], 'brevity_penalty': 1.0, 'length_ratio': 1.3529411764705883, 'translation_length': 69, 'reference_length': 51}

Prediction sentence trial 4

Reference sentence trial 3

{'bleu': 0.1486732977427379, 'precisions': [0.5476190476190477, 0.275, 0.21052631578947367, 0.16666666666666666], 'brevity_penalty': 0.551431257080004, 'length_ratio': 0.6268656716417911, 'translation_length': 42, 'reference_length': 67}

Prediction sentence trial 5

Reference sentence trial 4

{'bleu': 0.33498832882349805, 'precisions': [0.59375, 0.3548387096774194, 0.26666666666666666, 0.22413793103448276], 'brevity_penalty': 1.0, 'length_ratio': 1.3333333333333333, 'translation_length': 64, 'reference_length': 48}

Prediction sentence trial 6

Reference sentence trial 5

{'bleu': 0.5278131299382726, 'precisions': [0.8032786885245902, 0.6271186440677966, 0.49122807017543857, 0.38181818181818183], 'brevity_penalty': 0.952009440385274, 'length_ratio': 0.953125, 'translation_length': 61, 'reference_length': 64}

Prediction sentence trial 7

Reference sentence trial 6

{'bleu': 0.0, 'precisions': [0.296875, 0.06451612903225806, 0.0, 0.0], 'brevity_penalty': 1.0, 'length_ratio': 1.2549019607843137, 'translation_length': 64, 'reference_length': 51}

Prediction sentence trial 8

Reference sentence trial 7

{'bleu': 0.23101577886451632, 'precisions': [0.6122448979591837, 0.425531914893617, 0.35555555555555557, 0.3023255813953488], 'brevity_penalty': 0.5647181220077593, 'length_ratio': 0.6363636363636364, 'translation_length': 49, 'reference_length': 77}

Prediction sentence trial 9

Reference sentence trial 8
{'bleu': 0.11964747814499822, 'precisions': [0.2595419847328244, 0.16279069767441862, 0.08661417322834646, 0.056], 'brevity_penalty': 1.0, 'length_ratio': 3.638888888888889, 'translation_length': 131, 'reference_length': 36}

Prediction sentence trial 10
Reference sentence trial 9
{'bleu': 0.08234219463501491, 'precisions': [0.5454545454545454, 0.2641509433962264, 0.13725490196078433, 0.10204081632653061], 'brevity_penalty': 0.38850293851629, 'length_ratio': 0.514018691588785, 'translation_length': 55, 'reference_length': 107}

Round 2:
Prediction sentence trial 1
Reference sentence trial 2
{'bleu': 0.6336577611447871, 'precisions': [0.7777777777777778, 0.6285714285714286, 0.5882352941176471, 0.5606060606060606], 'brevity_penalty': 1.0, 'length_ratio': 1.1428571428571428, 'translation_length': 72, 'reference_length': 63}

Prediction sentence trial 2
Reference sentence trial 3
{'bleu': 0.1644925430446001, 'precisions': [0.41509433962264153, 0.21568627450980393, 0.1836734693877551, 0.14893617021276595], 'brevity_penalty': 0.7394217682762253, 'length_ratio': 0.7681159420289855, 'translation_length': 53, 'reference_length': 69}

Prediction sentence trial 3
Reference sentence trial 4
{'bleu': 0.11082305217917361, 'precisions': [0.5, 0.225, 0.15789473684210525, 0.1111111111111111], 'brevity_penalty': 0.5257880244257798, 'length_ratio': 0.6086956521739131, 'translation_length': 42, 'reference_length': 69}

Prediction sentence trial 4
Reference sentence trial 5
{'bleu': 0.29376862593961844, 'precisions': [0.72, 0.4166666666666667, 0.30434782608695654, 0.25], 'brevity_penalty': 0.7557837414557255, 'length_ratio': 0.78125, 'translation_length': 50, 'reference_length': 64}

Prediction sentence trial 5
Reference sentence trial 6
{'bleu': 0.47687713586922476, 'precisions': [0.7121212121212122, 0.546875, 0.41935483870967744, 0.31666666666666665], 'brevity_penalty': 1.0, 'length_ratio': 1.0819672131147542, 'translation_length': 66, 'reference_length': 61}

Prediction sentence trial 6
Reference sentence trial 7

{'bleu': 0.0, 'precisions': [0.3333333333333333, 0.057692307692307696, 0.0, 0.0], 'brevity_penalty': 0.830950389901292, 'length_ratio': 0.84375, 'translation_length': 54, 'reference_length': 64}

Prediction sentence trial 7
Reference sentence trial 8
{'bleu': 0.055694822295586206, 'precisions': [0.2077922077922078, 0.08, 0.0410958904109589, 0.014084507042253521], 'brevity_penalty': 1.0, 'length_ratio': 2.138888888888889, 'translation_length': 77, 'reference_length': 36}

Prediction sentence trial 8
Reference sentence trial 9
{'bleu': 0.1806945521043731, 'precisions': [0.847457627118644, 0.6666666666666666, 0.5272727272727272, 0.4716981132075472], 'brevity_penalty': 0.29513010586178823, 'length_ratio': 0.45038167938931295, 'translation_length': 59, 'reference_length': 131}

Prediction sentence trial 9
Reference sentence trial 10
{'bleu': 0.22025475316319115, 'precisions': [0.3515625, 0.23015873015873015, 0.1774193548387097, 0.16393442622950818], 'brevity_penalty': 1.0, 'length_ratio': 2.327272727272727, 'translation_length': 128, 'reference_length': 55}

Question 2:
Prediction sentence trial 2
Reference sentence trial 1
{'bleu': 0.19193032517483052, 'precisions': [0.4074074074074074, 0.2553191489361702, 0.14705882352941177, 0.08870967741935484], 'brevity_penalty': 1.0, 'length_ratio': 1.9585492227979275, 'translation_length': 378, 'reference_length': 193}

Prediction sentence trial 3
Reference sentence trial 2
{'bleu': 0.07098557646005438, 'precisions': [0.7878787878787878, 0.5846153846153846, 0.3984375, 0.25396825396825395], 'brevity_penalty': 0.1527751756017147, 'length_ratio': 0.3473684210526316, 'translation_length': 132, 'reference_length': 380}

Prediction sentence trial 4
Reference sentence trial 3
{'bleu': 0.3283178847439476, 'precisions': [0.5572139303482587, 0.36180904522613067, 0.27411167512690354, 0.21025641025641026], 'brevity_penalty': 1.0, 'length_ratio': 1.5, 'translation_length': 201, 'reference_length': 134}

Prediction sentence trial 5

Reference sentence trial 4
{'bleu': 0.3161098955348137, 'precisions': [0.743421052631579, 0.48, 0.35135135135135137, 0.2602739726027397], 'brevity_penalty': 0.7437482823436001, 'length_ratio': 0.7715736040609137, 'translation_length': 152, 'reference_length': 197}

Prediction sentence trial 6
Reference sentence trial 5
{'bleu': 0.3484457314007012, 'precisions': [0.6690140845070423, 0.45, 0.30434782608695654, 0.21323529411764705], 'brevity_penalty': 0.9319999339074818, 'length_ratio': 0.9342105263157895, 'translation_length': 142, 'reference_length': 152}

Prediction sentence trial 7
Reference sentence trial 6
{'bleu': 0.6170408616738879, 'precisions': [0.9083969465648855, 0.7829457364341085, 0.7007874015748031, 0.624], 'brevity_penalty': 0.826265032234987, 'length_ratio': 0.8397435897435898, 'translation_length': 131, 'reference_length': 156}

Prediction sentence trial 8
Reference sentence trial 7
{'bleu': 0.337044953033719, 'precisions': [0.4432624113475177, 0.36428571428571427, 0.302158273381295, 0.2644927536231884], 'brevity_penalty': 1.0, 'length_ratio': 2.028776978417266, 'translation_length': 282, 'reference_length': 139}

Prediction sentence trial 9
Reference sentence trial 8
{'bleu': 0.3507026049493344, 'precisions': [0.625, 0.4064748201438849, 0.286231884057971, 0.20802919708029197], 'brevity_penalty': 1.0, 'length_ratio': 1.033210332103321, 'translation_length': 280, 'reference_length': 271}

Prediction sentence trial 10
Reference sentence trial 9
{'bleu': 0.0943385732164411, 'precisions': [0.8556701030927835, 0.7684210526315789, 0.6881720430107527, 0.6153846153846154], 'brevity_penalty': 0.12986795007771892, 'length_ratio': 0.3288135593220339, 'translation_length': 97, 'reference_length': 295}

Round 2:
Prediction sentence trial 1
Reference sentence trial 2
{'bleu': 0.1452616543660978, 'precisions': [0.7979274611398963, 0.5026178010471204, 0.291005291005291, 0.17647058823529413], 'brevity_penalty': 0.3834487813876141, 'length_ratio': 0.5105820105820106, 'translation_length': 193, 'reference_length': 378}

Prediction sentence trial 2
Reference sentence trial 3

{'bleu': 0.15896739613424585, 'precisions': [0.2736842105263158, 0.20105820105820105, 0.1356382978723404, 0.0855614973262032], 'brevity_penalty': 1.0, 'length_ratio': 2.878787878787879, 'translation_length': 380, 'reference_length': 132}

Prediction sentence trial 3
Reference sentence trial 4
{'bleu': 0.30100712803532104, 'precisions': [0.835820895522388, 0.5454545454545454, 0.4153846153846154, 0.3203125], 'brevity_penalty': 0.6065306597126334, 'length_ratio': 0.6666666666666666, 'translation_length': 134, 'reference_length': 201}

Prediction sentence trial 4
Reference sentence trial 5
{'bleu': 0.32641987308966747, 'precisions': [0.5736040609137056, 0.36923076923076925, 0.2694300518134715, 0.19895287958115182], 'brevity_penalty': 1.0, 'length_ratio': 1.2960526315789473, 'translation_length': 197, 'reference_length': 152}

Prediction sentence trial 5
Reference sentence trial 6
{'bleu': 0.3487711121533964, 'precisions': [0.625, 0.42, 0.28378378378378377, 0.19863013698630136], 'brevity_penalty': 1.0, 'length_ratio': 1.0704225352112675, 'translation_length': 152, 'reference_length': 142}

Prediction sentence trial 6
Reference sentence trial 7
{'bleu': 0.6102483711255138, 'precisions': [0.7756410256410257, 0.6493506493506493, 0.5657894736842105, 0.48666666666666667], 'brevity_penalty': 1.0, 'length_ratio': 1.1223021582733812, 'translation_length': 156, 'reference_length': 139}

Prediction sentence trial 7
Reference sentence trial 8
{'bleu': 0.20938310933547807, 'precisions': [0.7913669064748201, 0.6058394160583942, 0.4666666666666667, 0.38345864661654133], 'brevity_penalty': 0.38688016395433344, 'length_ratio': 0.5129151291512916, 'translation_length': 139, 'reference_length': 271}

Prediction sentence trial 8
Reference sentence trial 9
{'bleu': 0.35064033233170594, 'precisions': [0.6457564575645757, 0.4200743494423792, 0.2958801498127341, 0.21509433962264152], 'brevity_penalty': 0.9673350765690054, 'length_ratio': 0.9678571428571429, 'translation_length': 271, 'reference_length': 280}

Prediction sentence trial 9
Reference sentence trial 10

{'bleu': 0.23378794921674637, 'precisions': [0.28135593220338984, 0.24914675767918087, 0.21993127147766323, 0.19377162629757785], 'brevity_penalty': 1.0, 'length_ratio': 3.0412371134020617, 'translation_length': 295, 'reference_length': 97}

Question 3:
Prediction sentence trial 2
Reference sentence trial 1
{'bleu': 0.14454497534644306, 'precisions': [0.25379609544468545, 0.1655773420479303, 0.11816192560175055, 0.08791208791208792], 'brevity_penalty': 1.0, 'length_ratio': 3.269503546099291, 'translation_length': 461, 'reference_length': 141}

Prediction sentence trial 3
Reference sentence trial 2
{'bleu': 0.0011067028042717356, 'precisions': [0.8666666666666667, 0.7931034482758621, 0.7321428571428571, 0.6666666666666666], 'brevity_penalty': 0.001454150593543583, 'length_ratio': 0.13274336283185842, 'translation_length': 60, 'reference_length': 452}

Prediction sentence trial 4
Reference sentence trial 3
{'bleu': 0.6039525029430881, 'precisions': [0.7252747252747253, 0.6292134831460674, 0.5632183908045977, 0.5176470588235295], 'brevity_penalty': 1.0, 'length_ratio': 1.1818181818181819, 'translation_length': 91, 'reference_length': 77}

Prediction sentence trial 5
Reference sentence trial 4
{'bleu': 0.5945947233510436, 'precisions': [0.8987341772151899, 0.7922077922077922, 0.72, 0.6712328767123288], 'brevity_penalty': 0.7763401191404025, 'length_ratio': 0.797979797979798, 'translation_length': 79, 'reference_length': 99}

Prediction sentence trial 6
Reference sentence trial 5
{'bleu': 0.16346352822589838, 'precisions': [0.26785714285714285, 0.19369369369369369, 0.13636363636363635, 0.10091743119266056], 'brevity_penalty': 1.0, 'length_ratio': 3.246376811594203, 'translation_length': 224, 'reference_length': 69}

Prediction sentence trial 7
Reference sentence trial 6
{'bleu': 0.30627625539015596, 'precisions': [0.43171806167400884, 0.334070796460177, 0.2733333333333333, 0.22321428571428573], 'brevity_penalty': 1.0, 'length_ratio': 2.0730593607305936, 'translation_length': 454, 'reference_length': 219}

Prediction sentence trial 8
Reference sentence trial 7

{'bleu': 0.0329046595226197, 'precisions': [0.9380530973451328, 0.7657657657657657, 0.6513761467889908, 0.5607476635514018], 'brevity_penalty': 0.045975528292226724, 'length_ratio': 0.24511930585683298, 'translation_length': 113, 'reference_length': 461}

Prediction sentence trial 9
Reference sentence trial 8
{'bleu': 0.2729468674318008, 'precisions': [0.6931818181818182, 0.4069767441860465, 0.2619047619047619, 0.1951219512195122], 'brevity_penalty': 0.7877012671239861, 'length_ratio': 0.8073394495412844, 'translation_length': 88, 'reference_length': 109}

Prediction sentence trial 10
Reference sentence trial 9
{'bleu': 0.07336793075427103, 'precisions': [0.7631578947368421, 0.4166666666666667, 0.2647058823529412, 0.125], 'brevity_penalty': 0.2290799498154876, 'length_ratio': 0.40425531914893614, 'translation_length': 38, 'reference_length': 94}

Round 2:
Prediction sentence trial 1
Reference sentence trial 2
{'bleu': 0.049591628779479036, 'precisions': [0.8297872340425532, 0.5467625899280576, 0.39416058394160586, 0.2962962962962963], 'brevity_penalty': 0.10336348255149484, 'length_ratio': 0.30585683297180044, 'translation_length': 141, 'reference_length': 461}

Prediction sentence trial 2
Reference sentence trial 3
{'bleu': 0.09654303063603843, 'precisions': [0.11504424778761062, 0.10222222222222223, 0.09151785714285714, 0.08071748878923767], 'brevity_penalty': 1.0, 'length_ratio': 7.533333333333333, 'translation_length': 452, 'reference_length': 60}

Prediction sentence trial 3
Reference sentence trial 4
{'bleu': 0.5988930142892133, 'precisions': [0.8571428571428571, 0.7466666666666667, 0.6712328767123288, 0.6197183098591549], 'brevity_penalty': 0.8337529180751805, 'length_ratio': 0.8461538461538461, 'translation_length': 77, 'reference_length': 91}

Prediction sentence trial 4
Reference sentence trial 5
{'bleu': 0.4072318096613296, 'precisions': [0.5757575757575758, 0.44329896907216493, 0.35789473684210527, 0.3010752688172043], 'brevity_penalty': 1.0, 'length_ratio': 1.434782608695652, 'translation_length': 99, 'reference_length': 69}

Prediction sentence trial 5
Reference sentence trial 6

{'bleu': 0.05317129644743472, 'precisions': [0.8115942028985508, 0.5671641791044776, 0.38461538461538464, 0.2698412698412698], 'brevity_penalty': 0.11373170793120879, 'length_ratio': 0.3150684931506849, 'translation_length': 69, 'reference_length': 219}

Prediction sentence trial 6
Reference sentence trial 7
{'bleu': 0.2081326032248073, 'precisions': [0.8904109589041096, 0.6820276497695853, 0.5581395348837209, 0.460093896713615], 'brevity_penalty': 0.33120332043601475, 'length_ratio': 0.4750542299349241, 'translation_length': 219, 'reference_length': 461}

Prediction sentence trial 7
Reference sentence trial 8
{'bleu': 0.15698481773809497, 'precisions': [0.21908893709327548, 0.16993464052287582, 0.1400437636761488, 0.11648351648351649], 'brevity_penalty': 1.0, 'length_ratio': 4.229357798165138, 'translation_length': 461, 'reference_length': 109}

Prediction sentence trial 8
Reference sentence trial 9
{'bleu': 0.2737431567428728, 'precisions': [0.5688073394495413, 0.3177570093457944, 0.2, 0.1553398058252427], 'brevity_penalty': 1.0, 'length_ratio': 1.1595744680851063, 'translation_length': 109, 'reference_length': 94}

Prediction sentence trial 9
Reference sentence trial 10
{'bleu': 0.12296675820332553, 'precisions': [0.30851063829787234, 0.16304347826086957, 0.1, 0.045454545454545456], 'brevity_penalty': 1.0, 'length_ratio': 2.473684210526316, 'translation_length': 94, 'reference_length': 38}

Question 4
Prediction sentence trial 2
Reference sentence trial 1
{'bleu': 0.37471387283259155, 'precisions': [0.7626262626262627, 0.5126903553299492, 0.35714285714285715, 0.2641025641025641], 'brevity_penalty': 0.8550753745473764, 'length_ratio': 0.8646288209606987, 'translation_length': 396, 'reference_length': 458}

Prediction sentence trial 3
Reference sentence trial 2
{'bleu': 0.2556473108386591, 'precisions': [0.7906137184115524, 0.48363636363636364, 0.31135531135531136, 0.2029520295202952], 'brevity_penalty': 0.6484223654723225, 'length_ratio': 0.6977329974811083, 'translation_length': 277, 'reference_length': 397}

Prediction sentence trial 4
Reference sentence trial 3

{'bleu': 0.25339764543777465, 'precisions': [0.5531914893617021, 0.3180428134556575, 0.18461538461538463, 0.12693498452012383], 'brevity_penalty': 1.0, 'length_ratio': 1.187725631768953, 'translation_length': 329, 'reference_length': 277}

Prediction sentence trial 5
Reference sentence trial 4
{'bleu': 0.3024256260910178, 'precisions': [0.7314487632508834, 0.45195729537366547, 0.2724014336917563, 0.18050541516245489], 'brevity_penalty': 0.8469803886316553, 'length_ratio': 0.8575757575757575, 'translation_length': 283, 'reference_length': 330}

Prediction sentence trial 6
Reference sentence trial 5
{'bleu': 0.283458500867465, 'precisions': [0.7067137809187279, 0.398576512455516, 0.2007168458781362, 0.11913357400722022], 'brevity_penalty': 0.9894552827603211, 'length_ratio': 0.9895104895104895, 'translation_length': 283, 'reference_length': 286}

Prediction sentence trial 7
Reference sentence trial 6
{'bleu': 0.5267827711990033, 'precisions': [0.7906976744186046, 0.64453125, 0.5354330708661418, 0.45634920634920634], 'brevity_penalty': 0.8867829403316152, 'length_ratio': 0.8927335640138409, 'translation_length': 258, 'reference_length': 289}

Prediction sentence trial 8
Reference sentence trial 7
{'bleu': 0.24049925488667992, 'precisions': [0.8300653594771242, 0.5629139072847682, 0.42953020134228187, 0.3741496598639456], 'brevity_penalty': 0.4594258240359267, 'length_ratio': 0.5625, 'translation_length': 153, 'reference_length': 272}

Prediction sentence trial 9
Reference sentence trial 8
{'bleu': 0.08238873744304588, 'precisions': [0.4473684210526316, 0.1875, 0.06363636363636363, 0.009259259259259259], 'brevity_penalty': 0.982609137827942, 'length_ratio': 0.9827586206896551, 'translation_length': 114, 'reference_length': 116}

Prediction sentence trial 10
Reference sentence trial 9
{'bleu': 0.15384721619965372, 'precisions': [0.284375, 0.1761006289308176, 0.11708860759493671, 0.09554140127388536], 'brevity_penalty': 1.0, 'length_ratio': 2.7586206896551726, 'translation_length': 320, 'reference_length': 116}

Round 2:
Prediction sentence trial 1
Reference sentence trial 2

{'bleu': 0.317809970412019, 'precisions': [0.6179039301310044, 0.38596491228070173, 0.2511013215859031, 0.17035398230088494], 'brevity_penalty': 1.0, 'length_ratio': 1.1536523929471032, 'translation_length': 458, 'reference_length': 397}

Prediction sentence trial 2
Reference sentence trial 3
{'bleu': 0.20257357369121565, 'precisions': [0.5088161209068011, 0.27341772151898736, 0.15267175572519084, 0.0792838874680307], 'brevity_penalty': 1.0, 'length_ratio': 1.4332129963898916, 'translation_length': 397, 'reference_length': 277}

Prediction sentence trial 3
Reference sentence trial 4
{'bleu': 0.22067549443793064, 'precisions': [0.6425992779783394, 0.3527272727272727, 0.19047619047619047, 0.11808118081180811], 'brevity_penalty': 0.8258552688925983, 'length_ratio': 0.8393939393939394, 'translation_length': 277, 'reference_length': 330}

Prediction sentence trial 4
Reference sentence trial 5
{'bleu': 0.24080544425147696, 'precisions': [0.593939393939394, 0.3384146341463415, 0.17484662576687116, 0.09567901234567901], 'brevity_penalty': 1.0, 'length_ratio': 1.1538461538461537, 'translation_length': 330, 'reference_length': 286}

Prediction sentence trial 5
Reference sentence trial 6
{'bleu': 0.28949816618991536, 'precisions': [0.7097902097902098, 0.4014084507042254, 0.20567375886524822, 0.125], 'brevity_penalty': 0.9895653125691847, 'length_ratio': 0.9896193771626297, 'translation_length': 286, 'reference_length': 289}

Prediction sentence trial 6
Reference sentence trial 7
{'bleu': 0.47235920601931153, 'precisions': [0.6782006920415224, 0.5226480836236934, 0.41403508771929826, 0.3392226148409894], 'brevity_penalty': 1.0, 'length_ratio': 1.0625, 'translation_length': 289, 'reference_length': 272}

Prediction sentence trial 7
Reference sentence trial 8
{'bleu': 0.13831240982842005, 'precisions': [0.3272058823529412, 0.17037037037037037, 0.09701492537313433, 0.06766917293233082], 'brevity_penalty': 1.0, 'length_ratio': 2.3448275862068964, 'translation_length': 272, 'reference_length': 116}

Prediction sentence trial 8
Reference sentence trial 9

{'bleu': 0.08315816816570401, 'precisions': [0.45689655172413796, 0.18421052631578946, 0.0625, 0.00909090909090909], 'brevity_penalty': 1.0, 'length_ratio': 1.0, 'translation_length': 116, 'reference_length': 116}

Prediction sentence trial 9
Reference sentence trial 10
{'bleu': 0.07436788854328015, 'precisions': [0.7844827586206896, 0.49122807017543857, 0.33035714285714285, 0.2727272727272727], 'brevity_penalty': 0.17228233081618782, 'length_ratio': 0.3625, 'translation_length': 116, 'reference_length': 320}

Question 5
Prediction sentence trial 2
Reference sentence trial 1
{'bleu': 0.29659157123521784, 'precisions': [0.6568627450980392, 0.3842364532019704, 0.2599009900990099, 0.18159203980099503], 'brevity_penalty': 0.8977684118033409, 'length_ratio': 0.9026548672566371, 'translation_length': 408, 'reference_length': 452}

Prediction sentence trial 3
Reference sentence trial 2
{'bleu': 0.04076220397836621, 'precisions': [1.0, 1.0, 1.0, 1.0], 'brevity_penalty': 0.04076220397836621, 'length_ratio': 0.23809523809523808, 'translation_length': 110, 'reference_length': 462}

Prediction sentence trial 4
Reference sentence trial 3
{'bleu': 0.3200299019670858, 'precisions': [0.639344262295082, 0.423728813559322, 0.3333333333333333, 0.2909090909090909], 'brevity_penalty': 0.7949244528369331, 'length_ratio': 0.8133333333333334, 'translation_length': 61, 'reference_length': 75}

Prediction sentence trial 5
Reference sentence trial 4
{'bleu': 0.33833638416926015, 'precisions': [0.5416666666666666, 0.35714285714285715, 0.27941176470588236, 0.24242424242424243], 'brevity_penalty': 1.0, 'length_ratio': 1.125, 'translation_length': 72, 'reference_length': 64}

Prediction sentence trial 6
Reference sentence trial 5
{'bleu': 0.10388222465752225, 'precisions': [0.17378048780487804, 0.11349693251533742, 0.08641975308641975, 0.06832298136645963], 'brevity_penalty': 1.0, 'length_ratio': 4.373333333333333, 'translation_length': 328, 'reference_length': 75}

Prediction sentence trial 7
Reference sentence trial 6

{'bleu': 0.18632551075086473, 'precisions': [0.745, 0.4292929292929293, 0.28061224489795916, 0.1958762886597938], 'brevity_penalty': 0.5117085777865424, 'length_ratio': 0.5988023952095808, 'translation_length': 200, 'reference_length': 334}

Prediction sentence trial 8
Reference sentence trial 7
{'bleu': 0.06254700455497074, 'precisions': [0.7361111111111112, 0.4142857142857143, 0.27941176470588236, 0.19696969696969696], 'brevity_penalty': 0.17377394345044514, 'length_ratio': 0.36363636363636365, 'translation_length': 72, 'reference_length': 198}

Prediction sentence trial 9
Reference sentence trial 8
{'bleu': 0.09974183361956798, 'precisions': [0.1657754010695187, 0.11290322580645161, 0.08108108108108109, 0.06521739130434782], 'brevity_penalty': 1.0, 'length_ratio': 4.921052631578948, 'translation_length': 374, 'reference_length': 76}

Prediction sentence trial 10
Reference sentence trial 9
{'bleu': 0.08515828898343998, 'precisions': [0.7777777777777778, 0.4370860927152318, 0.2483221476510067, 0.16326530612244897], 'brevity_penalty': 0.248538662326323, 'length_ratio': 0.4180327868852459, 'translation_length': 153, 'reference_length': 366}

Round 2:
Prediction sentence trial 1
Reference sentence trial 2
{'bleu': 0.2979901663069031, 'precisions': [0.5929203539823009, 0.3466666666666667, 0.234375, 0.16367713004484305], 'brevity_penalty': 1.0, 'length_ratio': 1.107843137254902, 'translation_length': 452, 'reference_length': 408}

Prediction sentence trial 2
Reference sentence trial 3
{'bleu': 0.2330673183926151, 'precisions': [0.23809523809523808, 0.23478260869565218, 0.2314410480349345, 0.22807017543859648], 'brevity_penalty': 1.0, 'length_ratio': 4.2, 'translation_length': 462, 'reference_length': 110}

Prediction sentence trial 3
Reference sentence trial 4
{'bleu': 0.3555521612345277, 'precisions': [0.56, 0.3698630136986301, 0.29577464788732394, 0.2608695652173913], 'brevity_penalty': 1.0, 'length_ratio': 1.171875, 'translation_length': 75, 'reference_length': 64}

Prediction sentence trial 4
Reference sentence trial 5

{'bleu': 0.31362279773765106, 'precisions': [0.640625, 0.3870967741935484, 0.3, 0.25862068965517243], 'brevity_penalty': 0.8420844271433824, 'length_ratio': 0.8533333333333334, 'translation_length': 64, 'reference_length': 75}

Prediction sentence trial 5
Reference sentence trial 6
{'bleu': 0.01308288706508791, 'precisions': [0.7333333333333333, 0.4520547945205479, 0.3380281690140845, 0.2608695652173913], 'brevity_penalty': 0.03163999375899008, 'length_ratio': 0.2245508982035928, 'translation_length': 75, 'reference_length': 334}

Prediction sentence trial 6
Reference sentence trial 7
{'bleu': 0.18964586824177238, 'precisions': [0.4281437125748503, 0.2319277108433735, 0.14242424242424243, 0.09146341463414634], 'brevity_penalty': 1.0, 'length_ratio': 1.6868686868686869, 'translation_length': 334, 'reference_length': 198}

Prediction sentence trial 7
Reference sentence trial 8
{'bleu': 0.13293052260303215, 'precisions': [0.2828282828282828, 0.15816326530612246, 0.10309278350515463, 0.06770833333333333], 'brevity_penalty': 1.0, 'length_ratio': 2.6052631578947367, 'translation_length': 198, 'reference_length': 76}

Prediction sentence trial 8
Reference sentence trial 9
{'bleu': 0.010051546493317295, 'precisions': [0.8157894736842105, 0.5675675675675675, 0.4166666666666667, 0.34285714285714286], 'brevity_penalty': 0.01982022036992181, 'length_ratio': 0.20320855614973263, 'translation_length': 76, 'reference_length': 374}

Prediction sentence trial 9
Reference sentence trial 10
{'bleu': 0.14157206948959816, 'precisions': [0.3251366120218579, 0.1813186813186813, 0.10220994475138122, 0.06666666666666667], 'brevity_penalty': 1.0, 'length_ratio': 2.392156862745098, 'translation_length': 366, 'reference_length': 153}

ROGUE Score

First question:
Prediction sentence trial 2
Reference sentence trial 1
{'rouge1': 0.8734939759036144, 'rouge2': 0.7592592592592593, 'rougeL': 0.8132530120481927, 'rougeLsum': 0.8132530120481927}

Prediction sentence trial 3
Reference sentence trial 2

{'rouge1': 0.6263157894736842, 'rouge2': 0.564516129032258, 'rougeL': 0.6052631578947368, 'rougeLsum': 0.6052631578947368}

Prediction sentence trial 4
Reference sentence trial 3
{'rouge1': 0.6477272727272727, 'rouge2': 0.5697674418604651, 'rougeL': 0.6136363636363636, 'rougeLsum': 0.6136363636363636}

Prediction sentence trial 5
Reference sentence trial 4
{'rouge1': 0.7727272727272727, 'rouge2': 0.6333333333333333, 'rougeL': 0.7077922077922079, 'rougeLsum': 0.7077922077922079}

Prediction sentence trial 6
Reference sentence trial 5
{'rouge1': 0.8555555555555555, 'rouge2': 0.7386363636363636, 'rougeL': 0.788888888888889, 'rougeLsum': 0.788888888888889}

Prediction sentence trial 7
Reference sentence trial 6
{'rouge1': 0.6212121212121212, 'rouge2': 0.520618556701031, 'rougeL': 0.5909090909090909, 'rougeLsum': 0.5909090909090909}

Prediction sentence trial 8
Reference sentence trial 7
{'rouge1': 0.6190476190476191, 'rouge2': 0.5365853658536586, 'rougeL': 0.5833333333333334, 'rougeLsum': 0.5833333333333334}

Prediction sentence trial 9
Reference sentence trial 8
{'rouge1': 0.5083333333333333, 'rouge2': 0.3444505194095134, 'rougeL': 0.5, 'rougeLsum': 0.5083333333333333}

Prediction sentence trial 10
Reference sentence trial 9
{'rouge1': 0.6596638655462185, 'rouge2': 0.5683760683760684, 'rougeL': 0.634453781512605, 'rougeLsum': 0.6596638655462185}

Round 2:

Prediction sentence trial 1
Reference sentence trial 2
{'rouge1': 0.7213200628601363, 'rouge2': 0.5925925925925926, 'rougeL': 0.6610790990047145, 'rougeLsum': 0.6610790990047145}

Prediction sentence trial 2

Reference sentence trial 3
{'rouge1': 0.41203007518796997, 'rouge2': 0.32767402376910015, 'rougeL': 0.3909774436090226, 'rougeLsum': 0.3909774436090226}

Prediction sentence trial 3
Reference sentence trial 4
{'rouge1': 0.5161483253588517, 'rouge2': 0.3638850889192886, 'rougeL': 0.42942583732057416, 'rougeLsum': 0.42942583732057416}

Prediction sentence trial 4
Reference sentence trial 5
{'rouge1': 0.5080213903743316, 'rouge2': 0.26666666666666666, 'rougeL': 0.4430863254392666, 'rougeLsum': 0.4430863254392666}

Prediction sentence trial 5
Reference sentence trial 6
{'rouge1': 0.5094017094017094, 'rouge2': 0.32954545454545453, 'rougeL': 0.44273504273504277, 'rougeLsum': 0.44273504273504277}

Prediction sentence trial 6
Reference sentence trial 7
{'rouge1': 0.3317384370015949, 'rouge2': 0.07944208611279563, 'rougeL': 0.3014354066985646, 'rougeLsum': 0.3014354066985646}

Prediction sentence trial 7
Reference sentence trial 8
{'rouge1': 0.21904761904761905, 'rouge2': 0.0921409214092141, 'rougeL': 0.18333333333333335, 'rougeLsum': 0.18333333333333335}

Prediction sentence trial 8
Reference sentence trial 9
{'rouge1': 0.4793478260869565, 'rouge2': 0.21388216303470542, 'rougeL': 0.34057971014492755, 'rougeLsum': 0.3489130434782609}

Prediction sentence trial 9
Reference sentence trial 10
{'rouge1': 0.4929971988795518, 'rouge2': 0.2941825199889716, 'rougeL': 0.4677871148459384, 'rougeLsum': 0.4929971988795518}

Question 2:
Prediction sentence trial 2
Reference sentence trial 1
{'rouge1': 0.7614770459081837, 'rouge2': 0.6623246492985972, 'rougeL': 0.6756487025948104, 'rougeLsum': 0.7375249500998005}

Prediction sentence trial 3
Reference sentence trial 2
{'rouge1': 0.7027334851936219, 'rouge2': 0.648741418764302, 'rougeL': 0.6343963553530751, 'rougeLsum': 0.6845102505694761}

Prediction sentence trial 4
Reference sentence trial 3
{'rouge1': 0.8136531365313653, 'rouge2': 0.6970260223048327, 'rougeL': 0.6992619926199262, 'rougeLsum': 0.7214022140221402}

Prediction sentence trial 5
Reference sentence trial 4
{'rouge1': 0.8288135593220339, 'rouge2': 0.7081911262798635, 'rougeL': 0.7203389830508474, 'rougeLsum': 0.8220338983050848}

Prediction sentence trial 6
Reference sentence trial 5
{'rouge1': 0.796, 'rouge2': 0.6935483870967742, 'rougeL': 0.7, 'rougeLsum': 0.744}

Prediction sentence trial 7
Reference sentence trial 6
{'rouge1': 0.840214932126697, 'rouge2': 0.7246376811594203, 'rougeL': 0.7994909502262443, 'rougeLsum': 0.7994909502262443}

Prediction sentence trial 8
Reference sentence trial 7
{'rouge1': 0.7835365853658537, 'rouge2': 0.7147239263803681, 'rougeL': 0.7195121951219512, 'rougeLsum': 0.725609756097561}

Prediction sentence trial 9
Reference sentence trial 8
{'rouge1': 0.8533333333333333, 'rouge2': 0.7299107142857143, 'rougeL': 0.7777777777777777, 'rougeLsum': 0.8488888888888889}

Prediction sentence trial 10
Reference sentence trial 9
{'rouge1': 0.6868512110726643, 'rouge2': 0.6533101045296167, 'rougeL': 0.6591695501730104, 'rougeLsum': 0.6799307958477508}

Round 2:
Prediction sentence trial 1
Reference sentence trial 2

{'rouge1': 0.6186199030510409, 'rouge2': 0.4123246492985972, 'rougeL': 0.38993441688052466, 'rougeLsum': 0.4597946963216425}

Prediction sentence trial 2
Reference sentence trial 3
{'rouge1': 0.41325980098309556, 'rouge2': 0.26638847758783146, 'rougeL': 0.3449226711425488, 'rougeLsum': 0.37453542740678575}

Prediction sentence trial 3
Reference sentence trial 4
{'rouge1': 0.6707959936742225, 'rouge2': 0.530359355638166, 'rougeL': 0.5564048497627834, 'rougeLsum': 0.6338956246705324}

Prediction sentence trial 4
Reference sentence trial 5
{'rouge1': 0.5192897497982244, 'rouge2': 0.3134542841746003, 'rougeL': 0.41081517352703795, 'rougeLsum': 0.4955609362389024}

Prediction sentence trial 5
Reference sentence trial 6
{'rouge1': 0.47457142857142864, 'rouge2': 0.30893300248138955, 'rougeL': 0.37857142857142856, 'rougeLsum': 0.3545714285714286}

Prediction sentence trial 6
Reference sentence trial 7
{'rouge1': 0.6380090497737557, 'rouge2': 0.4895833333333333, 'rougeL': 0.597285067873303, 'rougeLsum': 0.597285067873303}

Prediction sentence trial 7
Reference sentence trial 8
{'rouge1': 0.6078609096901779, 'rouge2': 0.4432953549517966, 'rougeL': 0.5168094924192485, 'rougeLsum': 0.5747363216875412}

Prediction sentence trial 8
Reference sentence trial 9
{'rouge1': 0.7650980392156863, 'rouge2': 0.5972576530612245, 'rougeL': 0.6895424836601307, 'rougeLsum': 0.7562091503267974}

Prediction sentence trial 9
Reference sentence trial 10
{'rouge1': 0.6868512110726643, 'rouge2': 0.6533101045296167, 'rougeL': 0.6591695501730104, 'rougeLsum': 0.6626297577854672}

Question 3:

Prediction sentence trial 2
Reference sentence trial 1
{'rouge1': 0.6823204419889503, 'rouge2': 0.6090573012939002, 'rougeL': 0.6160220994475138, 'rougeLsum': 0.6160220994475138}

Prediction sentence trial 3
Reference sentence trial 2
{'rouge1': 0.10593220338983049, 'rouge2': 0.09574468085106383, 'rougeL': 0.09533898305084744, 'rougeLsum': 0.09533898305084744}

Prediction sentence trial 4
Reference sentence trial 3
{'rouge1': 0.8697478991596639, 'rouge2': 0.8076923076923077, 'rougeL': 0.8529411764705883, 'rougeLsum': 0.8529411764705883}

Prediction sentence trial 5
Reference sentence trial 4
{'rouge1': 0.8697478991596639, 'rouge2': 0.8076923076923077, 'rougeL': 0.8529411764705883, 'rougeLsum': 0.8529411764705883}

Prediction sentence trial 6
Reference sentence trial 5
{'rouge1': 0.6872509960159363, 'rouge2': 0.6325301204819277, 'rougeL': 0.651394422310757, 'rougeLsum': 0.6832669322709163}

Prediction sentence trial 7
Reference sentence trial 6
{'rouge1': 0.7935483870967742, 'rouge2': 0.7378640776699029, 'rougeL': 0.7532258064516129, 'rougeLsum': 0.7612903225806451}

Prediction sentence trial 8
Reference sentence trial 7
{'rouge1': 0.6811175337186898, 'rouge2': 0.6450676982591876, 'rougeL': 0.6310211946050096, 'rougeLsum': 0.6310211946050096}

Prediction sentence trial 9
Reference sentence trial 8
{'rouge1': 0.5550847457627119, 'rouge2': 0.38285714285714284, 'rougeL': 0.49858757062146886, 'rougeLsum': 0.49858757062146886}

Prediction sentence trial 10
Reference sentence trial 9
{'rouge1': 0.6886792452830188, 'rouge2': 0.5865384615384616, 'rougeL': 0.6415094339622642, 'rougeLsum': 0.6415094339622642}

Round 2:
Prediction sentence trial 1
Reference sentence trial 2
{'rouge1': 0.18232044198895028, 'rouge2': 0.10905730129390018, 'rougeL': 0.11602209944751381, 'rougeLsum': 0.11602209944751381}

Prediction sentence trial 2
Reference sentence trial 3
{'rouge1': 0.33926553672316384, 'rouge2': 0.2743161094224924, 'rougeL': 0.32867231638418076, 'rougeLsum': 0.32867231638418076}

Prediction sentence trial 3
Reference sentence trial 4
{'rouge1': 0.581869111280876, 'rouge2': 0.43672456575682383, 'rougeL': 0.5650623885918004, 'rougeLsum': 0.5650623885918004}

Prediction sentence trial 4
Reference sentence trial 5
{'rouge1': 0.634453781512605, 'rouge2': 0.4951923076923077, 'rougeL': 0.6176470588235294, 'rougeLsum': 0.6176470588235294}

Prediction sentence trial 5
Reference sentence trial 6
{'rouge1': 0.39937220813714835, 'rouge2': 0.2938204430625729, 'rougeL': 0.3635156344319691, 'rougeLsum': 0.39538814439212844}

Prediction sentence trial 6
Reference sentence trial 7
{'rouge1': 0.4858560794044665, 'rouge2': 0.3628640776699029, 'rougeL': 0.4070719602977667, 'rougeLsum': 0.39739454094292803}

Prediction sentence trial 7
Reference sentence trial 8
{'rouge1': 0.1811175337186898, 'rouge2': 0.1450676982591876, 'rougeL': 0.13102119460500963, 'rougeLsum': 0.13102119460500963}

Prediction sentence trial 8
Reference sentence trial 9
{'rouge1': 0.6208742194469223, 'rouge2': 0.47697478991596637, 'rougeL': 0.5643770443056794, 'rougeLsum': 0.5643770443056794}

Prediction sentence trial 9
Reference sentence trial 10

{'rouge1': 0.4386792452830188, 'rouge2': 0.2865384615384615, 'rougeL': 0.3915094339622642, 'rougeLsum': 0.3915094339622642}

Question 4
Prediction sentence trial 2
Reference sentence trial 1
{'rouge1': 0.8520749665327978, 'rouge2': 0.7214765100671141, 'rougeL': 0.7302543507362784, 'rougeLsum': 0.8386880856760375}

Prediction sentence trial 3
Reference sentence trial 2
{'rouge1': 0.8141361256544503, 'rouge2': 0.6646234676007006, 'rougeL': 0.6989528795811518, 'rougeLsum': 0.7949389179755673}

Prediction sentence trial 4
Reference sentence trial 3
{'rouge1': 0.7876712328767124, 'rouge2': 0.637524557956778, 'rougeL': 0.6761252446183953, 'rougeLsum': 0.7700587084148728}

Prediction sentence trial 5
Reference sentence trial 4
{'rouge1': 0.8294797687861273, 'rouge2': 0.6721470019342359, 'rougeL': 0.710019267822736, 'rougeLsum': 0.8044315992292872}

Prediction sentence trial 6
Reference sentence trial 5
{'rouge1': 0.826808587407541, 'rouge2': 0.6397727272727273, 'rougeL': 0.6670575500631427, 'rougeLsum': 0.78116543388057}

Prediction sentence trial 7
Reference sentence trial 6
{'rouge1': 0.8628509719222461, 'rouge2': 0.7776572668112798, 'rougeL': 0.7980561555075594, 'rougeLsum': 0.8542116630669547}

Prediction sentence trial 8
Reference sentence trial 7
{'rouge1': 0.7582582582582582, 'rouge2': 0.6419939577039275, 'rougeL': 0.6471471471471472, 'rougeLsum': 0.6951951951951952}

Prediction sentence trial 9
Reference sentence trial 8
{'rouge1': 0.2893401015228426, 'rouge2': 0.12820512820512822, 'rougeL': 0.12690355329949238, 'rougeLsum': 0.12690355329949238}

Prediction sentence trial 10
Reference sentence trial 9
{'rouge1': 0.6866295264623956, 'rouge2': 0.5980392156862745, 'rougeL': 0.6337047353760445, 'rougeLsum': 0.6448467966573816}

Round 2:

Prediction sentence trial 1
Reference sentence trial 2
{'rouge1': 0.484150438230911, 'rouge2': 0.241084353204369, 'rougeL': 0.3434618979060897, 'rougeLsum': 0.4505569447601728}

Prediction sentence trial 2
Reference sentence trial 3
{'rouge1': 0.6202585746340421, 'rouge2': 0.3135596378134665, 'rougeL': 0.4642590020301314, 'rougeLsum': 0.5637354418207073}

Prediction sentence trial 3
Reference sentence trial 4
{'rouge1': 0.47134470226446745, 'rouge2': 0.24390753668018222, 'rougeL': 0.298574224210232, 'rougeLsum': 0.39446463516913616}

Prediction sentence trial 4
Reference sentence trial 5
{'rouge1': 0.5251319426991707, 'rouge2': 0.26305609284332687, 'rougeL': 0.3187149199966491, 'rougeLsum': 0.4150540336768032}

Prediction sentence trial 5
Reference sentence trial 6
{'rouge1': 0.5707699400645458, 'rouge2': 0.3252906976744186, 'rougeL': 0.3221300138312586, 'rougeLsum': 0.43623789764868603}

Prediction sentence trial 6
Reference sentence trial 7
{'rouge1': 0.5461843052555795, 'rouge2': 0.41558830129403845, 'rougeL': 0.48138948884089267, 'rougeLsum': 0.5397048236141109}

Prediction sentence trial 7
Reference sentence trial 8
{'rouge1': 0.25825825825825827, 'rouge2': 0.14199395770392748, 'rougeL': 0.14714714714714713, 'rougeLsum': 0.1741741741741742}

Prediction sentence trial 8
Reference sentence trial 9
{'rouge1': 0.47395548613822724, 'rouge2': 0.20757020757020758, 'rougeL': 0.26536509176103085, 'rougeLsum': 0.26536509176103085}

Prediction sentence trial 9
Reference sentence trial 10
{'rouge1': 0.3094365440062552, 'rouge2': 0.13440285204991087, 'rougeL': 0.23896789327078144, 'rougeLsum': 0.26125201583345553}

Question 5
Prediction sentence trial 2
Reference sentence trial 1
{'rouge1': 0.7968099861303745, 'rouge2': 0.6627260083449236, 'rougeL': 0.6678224687933426, 'rougeLsum': 0.7399445214979196}

Prediction sentence trial 3
Reference sentence trial 2
{'rouge1': 0.5977551020408164, 'rouge2': 0.5950811093668237, 'rougeL': 0.5977551020408164, 'rougeLsum': 0.5977551020408164}

Prediction sentence trial 4
Reference sentence trial 3
{'rouge1': 0.7577319587628866, 'rouge2': 0.6368421052631579, 'rougeL': 0.6855670103092784, 'rougeLsum': 0.6855670103092784}

Prediction sentence trial 5
Reference sentence trial 4
{'rouge1': 0.7577319587628866, 'rouge2': 0.6368421052631579, 'rougeL': 0.6855670103092784, 'rougeLsum': 0.6855670103092784}

Prediction sentence trial 6
Reference sentence trial 5
{'rouge1': 0.6227544910179641, 'rouge2': 0.5662650602409639, 'rougeL': 0.5898203592814372, 'rougeLsum': 0.6077844311377245}

Prediction sentence trial 7
Reference sentence trial 6
{'rouge1': 0.41195899772209565, 'rouge2': 0.24612255275870837, 'rougeL': 0.2980637813211845, 'rougeLsum': 0.384624145785877}

Prediction sentence trial 8
Reference sentence trial 7
{'rouge1': 0.6737089201877934, 'rouge2': 0.5710900473933649, 'rougeL': 0.6173708920187794, 'rougeLsum': 0.6455399061032864}

Prediction sentence trial 9
Reference sentence trial 8

{'rouge1': 0.6206434316353887, 'rouge2': 0.5673854447439353, 'rougeL': 0.5804289544235925, 'rougeLsum': 0.6045576407506702}

Prediction sentence trial 10
Reference sentence trial 9
{'rouge1': 0.7212189616252822, 'rouge2': 0.5975056689342404, 'rougeL': 0.626410835214447, 'rougeLsum': 0.698645598194131}
Round 2:
Prediction sentence trial 1
Reference sentence trial 2
{'rouge1': 0.42379411311450144, 'rouge2': 0.19551289359082516, 'rougeL': 0.26306056403143785, 'rougeLsum': 0.2727693018955155}

Prediction sentence trial 2
Reference sentence trial 3
{'rouge1': 0.25540216086434575, 'rouge2': 0.1358974358974359, 'rougeL': 0.25540216086434575, 'rougeLsum': 0.25540216086434575}

Prediction sentence trial 3
Reference sentence trial 4
{'rouge1': 0.33180603283696064, 'rouge2': 0.17530364372469637, 'rougeL': 0.2411225658648339, 'rougeLsum': 0.2411225658648339}

Prediction sentence trial 4
Reference sentence trial 5
{'rouge1': 0.7577319587628866, 'rouge2': 0.6368421052631579, 'rougeL': 0.6855670103092784, 'rougeLsum': 0.6855670103092784}

Prediction sentence trial 5
Reference sentence trial 6
{'rouge1': 0.27275449101796406, 'rouge2': 0.17737617135207495, 'rougeL': 0.23982035928143713, 'rougeLsum': 0.26377245508982033}

Prediction sentence trial 6
Reference sentence trial 7
{'rouge1': 0.7619589977220956, 'rouge2': 0.6350114416475973, 'rougeL': 0.6480637813211845, 'rougeLsum': 0.7277904328018223}

Prediction sentence trial 7
Reference sentence trial 8
{'rouge1': 0.3165660630449363, 'rouge2': 0.17635320528810178, 'rougeL': 0.2602280348759222, 'rougeLsum': 0.2649228705566734}

Prediction sentence trial 8

Reference sentence trial 9
{'rouge1': 0.12064343163538872, 'rouge2': 0.0673854447439353, 'rougeL': 0.08042895442359249, 'rougeLsum': 0.10723860589812331}

Prediction sentence trial 9
Reference sentence trial 10
{'rouge1': 0.22121896162528215, 'rouge2': 0.09750566893424037, 'rougeL': 0.12641083521444696, 'rougeLsum': 0.20090293453724606}

Appendix C: Prolog Rules for OpenAI's API and Rules Comparison between OpenAi's API and ChatGPT

Prolog code & Results Question 1:
4 rules in question 1

```
% DFS is better than BFS in certain situations Rule 1
dfs_better_than_bfs(Reason) :-
        memory_efficiency(Reason).
dfs_better_than_bfs(Reason) :-
        pathfinding(Reason).
dfs_better_than_bfs(Reason) :-
        space_complexity(Reason).
dfs_better_than_bfs(Reason) :-
        topological_sorting(Reason).

% Reasons why DFS might be preferred over BFS
memory_efficiency('DFS is more memory-efficient than BFS because it only need to keep track of the nodes along the current path from the root to the current node.').
Output: True
pathfinding('DFS is suitable for finding any solution or path from the start node to the goal node, as it tends to go deep into the search space quickly.').
Output: True
space_complexity('DFS has lower space complexity than BFS, especially when the tree/graph is sparse, since it does not require storing all nodes at the current depth.').
Output: True
topological_sorting('DFS is often used for topological sorting of directed acyclic graphs (DAGs), efficiently finding a topological ordering of nodes.').
Output: True
```

Prolog Question 2-1 rule:

```
% Facts
graph_representation(state_space_graph).
graph_representation(search_tree).

% Rules 1 rule
difference(Graph1, Graph2) :-
        graph_representation(Graph1),
        graph_representation(Graph2),
        not(Graph1 = Graph2).

% Query
?- difference(state_space_graph, search_tree).
Output: True
```

Prolog Question 3:
% Stochastic Hill Climbing: Exploration vs. Exploitation
augmentation_strategy(stochastic_hill_climbing, exploration).

% Random Restart Hill-Climbing:
augmentation_strategy(random_restart_hill_climbing, exploration).

% Stochastic Hill Climbing: Memory of Past States
memory_persistence(stochastic_hill_climbing, probabilistic).
history_consideration(stochastic_hill_climbing, simulated_annealing).

% Random Restart Hill-Climbing: Memory of Past States
memory_persistence(random_restart_hill_climbing, none).
history_consideration(random_restart_hill_climbing, none).

% Stochastic Hill Climbing: Convergence
convergence(stochastic_hill_climbing, gradual).

% Random Restart Hill-Climbing: Convergence
convergence(random_restart_hill_climbing, prevent_premature_convergence).

% Stochastic Hill Climbing: Global Optimum
global_optimum(stochastic_hill_climbing, probabilistic).
randomness_impact(stochastic_hill_climbing, degree_of_randomness).

% Random Restart Hill-Climbing: Global Optimum
global_optimum(random_restart_hill_climbing, increased_likelihood).

% Summary
summary(random_restart_hill_climbing,
"Random Restart Hill-Climbing addresses local optima by periodically restarting the search from different initial states. Both strategies aim to enhance the effectiveness of the basic hill-climbing algorithm in finding optimal solutions.").

Prolog Question 4- 4 Rules:
% Facts
tree_structured_csp(tree).
non_tree_structured_csp(non_tree).
efficient_algorithm(backtracking_search).
efficient_algorithm(dynamic_programming).
property(arc_consistency).
property(parallelization).
property(problem_decomposition).
property(consistency_check).

```
% Rules
transform_to_tree_structured_csp(non_tree) :-
    retract(non_tree_structured_csp(non_tree)),
    asserta(tree_structured_csp(tree)).

apply_algorithm(tree, Algorithm) :-
    efficient_algorithm(Algorithm),
    write('Applying '), write(Algorithm), write(' algorithm to tree-structured CSP'), nl.

apply_property(tree, Property) :-
    property(Property),
    write('Utilizing '), write(Property), write(' in tree-structured CSP'), nl.

translate_into_prolog :-
    transform_to_tree_structured_csp(non_tree),
    apply_algorithm(tree, backtracking_search),
    apply_algorithm(tree, dynamic_programming),
    apply_property(tree, arc_consistency),
    apply_property(tree, parallelization),
    apply_property(tree, problem_decomposition),
    apply_property(tree, consistency_check).
```

Prolog Question 5- 6 rules:

```
admissible_heuristic(h1).
admissible_heuristic(h2).

heuristic_evaluation_function(max(H1, H2)) :-
    admissible_heuristic(H1),
    admissible_heuristic(H2).

% Check if a heuristic is admissible
is_admissible(Heuristic) :-
    admissible_heuristic(Heuristic).

% Check if a heuristic dominates another heuristic
dominates(H1, H2) :-
    admissible_heuristic(H1),
    admissible_heuristic(H2),
    H1 >= H2.

% Combine and choose the maximum heuristic
max_heuristic(H1, H2, Max) :-
    H1 >= H2,
    Max is H1.
```

```
max_heuristic(H1, H2, Max) :-
    H1 < H2,
    Max is H2.

% Check if the combined heuristic is admissible
is_combined_heuristic_admissible(H1, H2, Max) :-
    admissible_heuristic(H1),
    admissible_heuristic(H2),
    max_heuristic(H1, H2, Max).

% Example predicate calls
?- heuristic_evaluation_function(max(h1, h2)).
?- is_admissible(h1).
?- is_admissible(h2).
?- dominates(h1, h2).
```

ChatGPT Prolog code Analysis

ChatGPT Prolog code Analysis

Question 1:

```
% Rule: dfs_better_than_bfs(+ProblemCharacteristics)
% Determines if DFS is better than BFS based on problem characteristics.
dfs_better_than_bfs(memory_usage, dfs).
dfs_better_than_bfs(solution_depth, dfs).
dfs_better_than_bfs(disconnected_graphs, dfs).
dfs_better_than_bfs(branching_factor(high), dfs).
dfs_better_than_bfs(path_length, dfs).
dfs_better_than_bfs(topological_sorting(DAG), dfs).

% Rule: bfs_better_than_dfs(+ProblemCharacteristics)
% Determines if BFS is better than DFS based on problem characteristics.
bfs_better_than_dfs(branching_factor(low), bfs).
bfs_better_than_dfs(topological_sorting(unweighted), bfs).

% Example Usage:
% Query: dfs_better_than_bfs(memory_usage, Strategy).
% Query: bfs_better_than_dfs(branching_factor(low), Strategy).
```

Question 2:

```
% Facts representing states and transitions in a simplified state space graph
state(a).
state(b).
state(c).
state(d).
```

```prolog
transition(a, b).
transition(b, c).
transition(b, d).

% Rule to check if a state is reachable in the state space graph
reachable(State) :- state(State).
reachable(NextState) :- transition(CurrentState, NextState), reachable(CurrentState).

% Facts representing nodes and edges in a simplified search tree
node(root).
node(a, root).
node(b, a).
node(c, b).
node(d, b).

edge(root, a).
edge(a, b).
edge(b, c).
edge(b, d).

% Rule to check if a node is part of the search tree
part_of_tree(Node) :- node(Node).
part_of_tree(Child) :- edge(Parent, Child), part_of_tree(Parent).

% Rule to check if a state is explored during a search
explored(State) :- part_of_tree(Node), state_at_node(State, Node).

% Rule to check if a state is associated with a specific node in the search tree
state_at_node(State, Node) :- node(Node, State).
state_at_node(State, Node) :- edge(Parent, Node), state_at_node(State, Parent).

% Example Usage:
% Query: reachable(c).
% Query: explored(c).
```

Question 3:

```prolog
% Stochastic Hill Climbing

% Rule: stochastic_hill_climbing(+CurrentState, -NextState)
% Stochastic Hill Climbing explores neighbors with a probability distribution.
stochastic_hill_climbing(CurrentState, NextState) :-
        neighbor_with_probability(CurrentState, NextState).

% Rule: neighbor_with_probability(+CurrentState, -NextState)
```

```
% Chooses a neighbor with some probability distribution.
neighbor_with_probability(CurrentState, NextState) :-
	% Define your probability distribution logic here.
	% For simplicity, let's say it randomly selects a neighbor.
	random_neighbor(CurrentState, NextState).

% Random Restart Hill Climbing

% Rule: random_restart_hill_climbing(+InitialStates, -BestState)
% Random Restart Hill Climbing restarts the search from multiple initial states.
random_restart_hill_climbing(InitialStates, BestState) :-
	findall(Result, (
	member(InitialState, InitialStates),
	hill_climbing(InitialState, Result)
	), Results),
	best_result(Results, BestState).

% Rule: best_result(+Results, -BestResult)
% Finds the best result among a list of results.
best_result([Result], Result).
best_result([Result | Rest], BestResult) :-
	best_result(Rest, RestBest),
	better_result(Result, RestBest, BestResult).

% Rule: better_result(+Result1, +Result2, -BetterResult)
% Compares two results and returns the better one.
% Modify this based on your optimization problem.
better_result(Result1, Result2, BetterResult) :-
	% For simplicity, let's say the result with a higher value is better.
	value(Result1, Value1),
	value(Result2, Value2),
	(Value1 >= Value2 -> BetterResult = Result1 ; BetterResult = Result2).

% Utility predicates (replace with your problem-specific details)
random_neighbor(CurrentState, NextState) :-
	% Define how a random neighbor is selected based on the current state.
	% For simplicity, let's assume a random choice.
	member(X, CurrentState),
	random_between(1, 10, NewX),
	replace(X, NewX, CurrentState, NextState).

hill_climbing(CurrentState, Result) :-
	% Define your hill climbing algorithm based on the problem.
	% For simplicity, let's say it explores neighbors in a deterministic way.
```

```
	deterministic_neighbor(CurrentState, Result).

deterministic_neighbor(CurrentState, NextState) :-
	% Define how a deterministic neighbor is selected based on the current state.
	% For simplicity, let's assume a deterministic choice.
	member(X, CurrentState),
	X1 is X + 1,
	replace(X, X1, CurrentState, NextState).

value(State, Value) :-
	% Define how the value of a state is calculated based on the problem.
	% For simplicity, let's assume a simple value function.
	sum(State, Value).

sum(List, Sum) :- sum(List, 0, Sum).
sum([], Acc, Acc).
sum([X | Xs], Acc, Sum) :- NewAcc is Acc + X, sum(Xs, NewAcc, Sum).

replace(_, _, [], []).
replace(Old, New, [Old | T], [New | T]).
replace(Old, New, [H | T], [H | NewT]) :- Old \= H, replace(Old, New, T, NewT).

% Example Usage:
% Query: stochastic_hill_climbing([1, 2, 3], NextState).
% Query: random_restart_hill_climbing([[1, 2, 3], [4, 5, 6]], BestState).
```

Question 4:

```
% Reasons for preferring tree-structured CSPs over general graph-structured CSPs

% 1. Simplicity of Search
reason(simplicity_of_search, tree_structured) :-
	write('Tree-structured CSPs allow for simpler and more efficient search algorithms.'), nl,
	write('Algorithms like depth-first search work well on tree-structured graphs.'), nl,
	write('Tree-structured CSPs often enable more straightforward backtracking and pruning strategies during the search process.'), nl.

% 2. Reduction in Complexity
reason(reduction_in_complexity, tree_structured) :-
	write('Transforming a general graph-structured CSP into a tree-structured one can simplify the representation and understanding of the problem.'), nl,
	write('This reduction in complexity can make it easier to design and implement efficient algorithms.'), nl.

% 3. Improved Computational Efficiency
```

reason(improved_computational_efficiency, tree_structured) :-
	write('Tree-structured CSPs may enable the use of more efficient inference techniques, such as constraint propagation and variable elimination.'), nl,
	write('Algorithms designed for tree-structured CSPs can take advantage of the hierarchical structure to reduce the search space and speed up the solving process.'), nl.

% 4. Parallelization Opportunities
reason(parallelization_opportunities, tree_structured) :-
	write('Tree-structured CSPs may offer opportunities for parallelization, as subtrees can be explored independently.'), nl,
	write('Parallel processing can lead to faster solutions, especially when solving large and complex CSPs.'), nl.

% Example usage:
% ?- Reason(simplicity_of_search, tree_structured).
% Output:
% Tree-structured CSPs allow for simpler and more efficient search algorithms.
% Algorithms like depth-first search work well on tree-structured graphs.
% Tree-structured CSPs often enable more straightforward backtracking and pruning strategies during the search process.

Question 5:
% Facts representing admissible heuristics h1 and h2
admissible_heuristic(h1).
admissible_heuristic(h2).

% Rule: max_heuristic(+State, -MaxHeuristic)
% Calculates the maximum admissible heuristic value for a given state.
max_heuristic(State, MaxHeuristic) :-
	findall(HeuristicValue, (admissible_heuristic(Heuristic), calculate_heuristic(Heuristic, State, HeuristicValue)), HeuristicValues),
	max_list(HeuristicValues, MaxHeuristic).

% Rule: calculate_heuristic(+Heuristic, +State, -HeuristicValue)
% Calculates the admissible heuristic value for a given state using a specific heuristic.
calculate_heuristic(h1, State, HeuristicValue) :-
	% Define the calculation for h1 based on the state.
	% For simplicity, let's assume a placeholder calculation.
	HeuristicValue is State * 2.

calculate_heuristic(h2, State, HeuristicValue) :-
	% Define the calculation for h2 based on the state.
	% For simplicity, let's assume a different placeholder calculation.
	HeuristicValue is State * State.

```
% Example Usage:
% Query: max_heuristic(10, MaxHeuristic).
% This query calculates the maximum admissible heuristic value for the state 10.

% Utility predicate to find the maximum value in a list
max_list([H], H).
max_list([H | T], Max) :-
        max_list(T, RestMax),
        (H >= RestMax -> Max = H ; Max = RestMax).
```

Results for Prolog API

Q1: 4 rules

Q2: 1 rule

Q3: 11 facts 0 rules

Q4:4 Rules

Q5: 6 rules

ChatGPT Rules Prolog on inspecting the LLM model of ChatGPT

Question 1: 2 rules

Question 2: 7 rules

Question 3: 10 rules

Question 4: 4 rules

Question 5: 3 rules