

ANALYSIS AND SURVEY OF GENERATIVE AI TOOL (CHATGPT) IN
COMPUTING EDUCATION

Savitha Rachuri

A Capstone Project (or Thesis) Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Department of Information Systems and Operations Management

University of North Carolina Wilmington

2023

Approved by

Advisory Committee

Dr. Ahmed ElSaid

Dr. Yao Shi

Dr. Karl Ricanek, Chair

Accepted By

Dean, Graduate School
TABLE OF CONTENTS

	Page
Abstract.....	6
Chapter 1: Introduction.....	6
1.1 Generative AI: ChatGPT.....	6
1.2 How did ChatGPT come into existence?	7
1.3 Can ChatGPT understand like humans?	7
1.4 Background and Motivation	8
1.5 Research Questions.....	8
Chapter 2: Literature Review	10
2.1 Probabilistic Language Models.....	10
2.2 Evolution of Neural Networks	10
2.2 Recurrent Neural Networks	12
2.3 Neural Machine Translation with Alignment	14
2.4 Transformer Model	16
2.5 Reinforcement Learning by Human Feedback (RLHF)	20
2.6 Large Language Model.....	11
2.7 Transformer Model	12
Chapter 3: Methodology	23
3.1 MIS 555 Database Management Systems.....	23
3.2 CSC 242 Computer Organization	23
3.3 Types of questions in courses	23
3.4 Crafting the Prompts.....	24
3.5 Workflow of the Methodology	25
Chapter 4: Preliminary Experiments.....	28
4.1 Experiment 1(MIS 555)	31
4.2 Experiment 2 (MIS 555)	31
4.3 Experiment 3(MIS555)	32
4.4 Experiment 1(CSC242)	34
4.5 Experiment 2(CSC242)	35
4.6 Experiment 3(CSC242)	37
4.7 Experiment 4(CSC242)	40
Chapter 5: Analysis and Results	42
5.1 How to rate Accuracy and Completeness?	43
5.2 Number of questions considered.....	46
5.3 Statistics of our Dataset	48
5.4 Correlation between accuracy scores of difficulty levels	51

Chapter 6: Conclusions and Future Work.....	55
References.....	56
Appendixes	
A. GitHub URL.....	57
Tables	
3.1 Accuracy Rating.....	25
3.2 Completeness Rating	26
5.1 Metadata of the curated Dataset.....	44
Figures	
2.1 Simple Neural Network	11
2.2 Evolution of Neural Networks	11
2.3 Example of image captioning vector-to-sequence model.....	12
2.4 Example of movie review sentiment analysis sequence-to-vector model.....	12
2.5 Example of language translation English to Spanish sequence-to-sequence model.....	13
2.6 Visual of Google Translate application where language translation is in play.....	13
2.7 Visual representation of input passing in RNN Encoder	13
2.8 Visual Representation of RNN Encoder-Decoder Model.....	14
2.9 High-level Transformer Encoder working.....	15
2.10 Example for input word embedding	16
2.11 Example to explain positional encoding	16
2.12 Transformer Encoder Architecture	17
2.13 Transformer Decoder Architecture	18
2.14 Transformer Encoder and Decoder Architecture	19
2.15 Reinforcement Learning	20
3.1 Workflow of Methodology	24
4.1 SQL Question 1.....	27
4.2 SQL Question 2	28
4.3 SQL Question 3.....	28
4.4 SQL Question 3 (continued-part1).....	29
4.5 SQL Question 3 (continued-part2).....	29
4.6 SQL Question 4.....	30
4.7 SQL Question 4 (contd.).....	31
4.8 SQL Question 5.....	32
4.9 SQL Question 5 (continued).....	33
4.10 ChatGPT Jan 30 version	33
4.11 ChatGPT Feb 13 version	34

4.12	CSC242 Question 1.....	35
4.13	CSC242 Question 2.....	36
4.14	LC3 instructions.....	37
4.15	CSC242 Question 3.....	38
4.16	CSC242 Question 3 (continued).....	39
4.17	CSC242 Question 4.....	40
5.1	Count of the questions based on difficulty type.....	49
5.2	Percentage Accuracy Calculation.	50
5.3	Percentage Completeness Calculation	50
5.4	Descriptive statistics for the whole data	51
5.5	Descriptive statistics based on the difficulty type	51
5.6	Stacked Accuracy Plot.....	52
5.7	Stacked Completeness Plot	52
5.8	Kruskal-Wallis Test	54
5.9	Mann-Whitney Test between Easy and Moderate	55
5.10	Mann-Whitney Test between Moderate and Hard.....	55
5.11	Mann-Whitney Test between Easy and Hard	55

Table	Page
3.1 Accuracy Rating.....	25
3.2 Completeness Rating	26
5.1 Metadata of the curated Dataset.....	44

Figure	Page
1. Simple Neural Network	11
2. Evolution of Neural Networks.....	11
3. Example of image captioning vector-to-sequence model.....	12
4. Example of movie review sentiment analysis sequence-to-vector model.....	12
5. Example of language translation English to Spanish sequence-to-sequence model..	13
6. Visual of Google Translate application where language translation is in play.....	13
7. Visual representation of input passing in RNN Encoder.....	13
8. Visual representation of RNN Encoder-Decoder Model.....	14
9. High-level Transformer Encoder working.....	15
10. Example for input word embedding	16
11. Example to explain positional encoding.....	16
12. Transformer Encoder Architecture	17
13. Transformer Decoder Architecture	18
14. Transformer Encoder and Decoder Architecture	19
15. Reinforcement Learning	23

16. SQL Question 1.....	30
17. SQL Question 2.....	31
18. SQL Question 3.....	31
19. SQL Question 3 (continued-part1)	32
20. SQL Question 3 (continued-part2)	32
21. SQL Question 4.....	33
22. SQL Question 4 (contd.)	34
23. SQL Question 5.....	35
24. SQL Question 5 (continued)	36
25. ChatGPT Jan 30 version.....	36
26. ChatGPT Feb 13 version.....	37
27. CSC242 Question 1.....	38
28. CSC242 Question 2.....	39
29. LC3 instructions.....	37
30. CSC242 Question 3.....	38
31. CSC242 Question (continued)	39
32. CSC242 Question 4.....	40
33. Count of the questions based on difficulty type.....	49
34. Percentage Accuracy calculation.	50
35. Percentage Completeness calculation	50
36. Descriptive statistics for the whole data	51
37. Descriptive statistics based on the difficulty type	51
38. Stacked Accuracy plot	52
39. Stacked Completeness Plot	52
40. Kruskal-Wallis Test	54
41. Mann-Whitney Test between Easy and Moderate	55
42. Mann-Whitney Test between Moderate and Hard.....	55
43. Mann-Whitney Test between Easy and Hard	55

ABSTRACT

Analysis and survey of Generative AI tool (ChatGPT) in Computing Education. Rachuri, Savitha, 2023. Capstone Paper, University of North Carolina Wilmington.

Since the origin of AI, the most talked about application of AI is how can Computers understand and process information like humans. Generative AI is one such Application of AI, which involves generating novel content in the form of text, audio, Images or videos, learning from extensive data fed to the models. One such application That uses the generative AI concept by using large language models that have come to Existence in recent days is OpenAI's ChatGPT. ChatGPT is trained on massive amounts of textual data, which allows the model employed to learn from the data and perform various kinds of text generation tasks that generate text which appear to be human written based on the prompt provided by users; These text generation tasks typically are, question answer, descriptive, text completion. Since its release in November 2022, it has created much buzz in the software industry. Learned experts in the industry have expressed diverse opinions on its impact on the education sector and the integrity of the Data it generates. In this project, I will research the effectiveness of the tool for Computing Education. I would be focusing on how it can be used to support the user, typically A student/professor. To validate this study, I will be surveying some of the questions in one of the undergraduate courses, CSC242 (Computer Organization), and a graduate course MIS555 (Database Management Systems) at the university, to understand how far ChatGPT can be used as a supporting tool for the students in the context of Computing Education.

Keywords: *Generative AI, ChatGPT, Large Language Models*

CHAPTER 1: INTRODUCTION

1.1 Generative AI: ChatGPT

Generative AI is a part of Artificial Intelligence where new output is generated as a response based on the input data they are trained on. Generative AI generates new content in the forms of images, text, music, and more, by learning and identifying patterns, and with this knowledge gained, generates new and unique outcomes. Generative AI produces intelligent responses that are exceptionally realistic and complex, making it a profitable tool for many industries such as education, finance, gaming, etc. One such generative AI tool that was recently developed and released is ChatGPT. ChatGPT, a large language model developed by OpenAI, garnered over 1 million subscribers within one week of its release. It has been the most talked about advancement in technology.

ChatGPT, short for Chat Generative Pre-Trained Transformer, is a model OpenAI trains to carry on a dialogue. It can respond to user input in a conversation and perform numerous language tasks, including text summarization and translation. The answers provided by it were astonishingly human-like. Besides answering general questions, it responds with code in several programming languages and solves math problems. Because of the tool's unique ability and potential, it has already been adopted by various companies such as Microsoft and Stripe to get ahead. In my project, I investigate the area of Computing Education to understand the extent to which tools like ChatGPT or other generative AI tools released in the future would affect.

1.2 How did ChatGPT come into existence?

In 2017, the "Attention is all you need" [1] paper introduced a network architecture called a transformer. This transformer network architecture is solely based on the attention mechanism. It took much less time to train and outperform the best models back then

regarding speed and performance. It quickly replaced Recurrent Neural Networks and Convolution Neural Networks in several deep learning tasks. 2018 the first Generative Pre-trained (GPT) transformer model was introduced. Within just two years, several state-of-the-art Transformer models were created, and the number of parameters in those models kept growing exponentially. The conversational format of the ChatGPT allows users to ask follow-up questions, reason with it, admit errors, and dismiss misguided requests [2].

1.3 Can ChatGPT understand like humans?

ChatGPT or other pre-trained models were a breakthrough in the field of NLP. The advent of such pre-trained models has become foundational in the NLP field; in other words, it has made it possible for many NLP tasks to be performed just by using the pre-trained models and fine-tuning them. ChatGPT is still a preview in progress. OpenAI clearly mentions this in its policy on its website [3], that it cannot be 100 percent reliable. The primary issue expressed by the CEO of OpenAI himself with regards to the responses given by ChatGPT is that, even though it provides factually accurate information on a lot of things, it is confident and wrong a significant fraction of time, and this can, in turn, mislead the user to believe that all the information given by it is always correct. To confuse the output of a large language model for accurate knowledge is not entirely right because the extent that the answer is ever correct is because you already know the answer. It almost always gives information that is commonly accepted facts. It is essential to understand that the model is fed with human-written text, and the responses given are helpful for the ones we expect in the output; they can be used to remind you of the concepts you have already learned. Besides this, the most important question of all time in the NLP realm remains, “Do language models understand the meaning of the words?”.

Although the pre-trained models have significantly improved the performance of NLP tasks, the critical challenge is interpretability [4]. No matter how many parameters there are, it is unclear whether it understands definitions and abstract concepts. In other words, the processes of decision-making by the models are dark and unknown to humans. For Example, in areas like healthcare and law, it is necessary to clearly understand the root cause of every outcome produced by the model. Therefore, there's still a lot of work to do for the robustness and truthfulness of the tool.

Another issue with most NLP models is getting trained on adversarial data (the kind of data that confuses the neural network), resulting in the misclassification of the given input. ChatGPT, in its early releases, had issues regarding racial and gender bias caused by such adversarial. After pointing these issues out to the users, the developers made significant changes by filtering these terms in the input data that triggers this kind of behavior in the model. As mentioned earlier, the tool is still in the testing stage and has a long way to go. But it would be interesting to see how these issues are navigated in the future.

1.4 Background and Motivation

Thus far in Spring 2023, UNCW has received 48 student plagiarism complaints. Out of the 48 complaints, 15 were reported to use AI in some capacity (31.3% of total complaints). It was observed that a few students from the undergraduate course CSC 242 had submitted the answers provided by ChatGPT to some of the homework assignment questions. These homework questions were mainly the LC3 binary instruction programming. As ChatGPT provides code on LC3 in an assembly programming language by default, students using this without prior knowledge of the workings of the tool allowed us to capture the issue. All these scenarios are the motivation to see how far one can rely on the responses of ChatGPT in the context of Computing Education.

1.5 Research Questions

Using ChatGPT, the focus of my project revolves around the following research questions.

Depending on the survey conducted, the analysis centers on the three chief functional attributes of the ChatGPT, namely:

- How crafting prompts makes a difference in the response provided by ChatGPT?

I am analyzing the difference in responses based on the context (prompt).

- How accurate are the results of ChatGPT?

I will explore the efficacy of the responses from ChatGPT. Coming up with a rating scale for the ambiguous answer associated with natural language. This measure must include an estimate of the response's correctness, Completeness, and accuracy.

- How to measure the performance or accuracy of the tool in computing education?

Metric to measure the tool's performance in the above context, considering the responses from the device as a classification problem and comparing the performance over various difficulty levels of the questions. [4]

CHAPTER 2: LITERATURE REVIEW

2.1 Probabilistic Language Models

Large Language Models belong to a category of machine learning called Natural Language Processing, which is the idea behind making computers capable of reading and understanding text like humans. Before the emergence of Neural Networks, language models that were developed earlier were built using probabilistic methods. In a simple probabilistic language model, in the context of a sequence of words, calculating the probability of the word appearing next depends on the conditional probability of it occurring next to the previous word. This concept is called the Markov assumption. The main drawback of this approach is that the context to predict the next word cannot be decided just by considering the preceding one to two terms. Also, if we were to consider more than five preceding words, the conditional probability computation would turn complex. And it scales poorly too. Probabilistic language models cannot generate huge texts or essays. Therefore, there needs to be an approximate function where there is no need for complex probability computation. Neural Networks were then introduced in language models.

2.2 Evolution of Neural Networks

Neural Networks, unlike other approximate function methods such as Fourier Transform, etc., have a unique ability to produce an approximate function where you provide input, giving an approximated output without requiring the knowledge of the input function. Figure 2.1 represents a simple neural network. A typical neural network consists of an input layer, one or more hidden layers, and an output layer.

A simple neural network

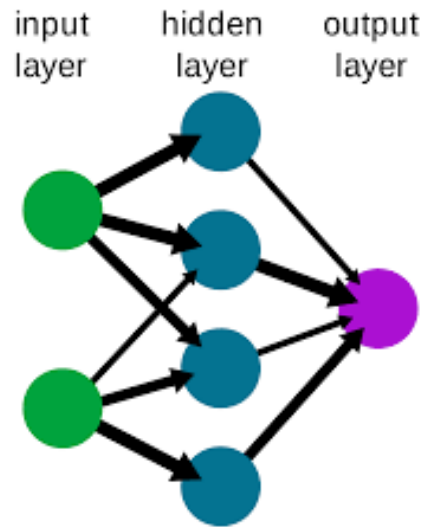


Figure 2.1: Simple Neural Network

The inputs from the input layer are assigned random weights, passed through an activation function in the hidden layer, and then passed through an output layer producing an approximate value. The error is calculated between the desired output value and the approximated value. The error usually is either root mean square or mean square error. With a phenomenon called backpropagation, a method called Gradient Descent, the weights that were assigned initially are updated over and over until the error is minimized.

The basis of the mechanism behind Generative AI tools comes from Neural Networks. In 1958, a single-neuron perceptron was discovered, a single-layer neural network that is a feed-forward network. The concept of backpropagation was foundational in unveiling the real potential of deep neural networks. This led to the development of CNNs (Convolution Neural Networks). The first ever CNN framework, LeNet-5, has 7 layers of neural networks. CNNs were mainly used in image classification. The Computer Vision area has greatly benefitted due to CNNs. It was only after the breakthrough of RNNs (Recurrent Neural Networks) that the field of Natural Language Processing, which is the

study of giving the computer the ability to interpret, manipulate, and comprehend human language, started to gain momentum. The advent of RNNs has seen a significant development that led to the development of large language neural network models, with several hundreds of layers and parameters in text input models or large language models. With the introduction of GANs in 2014, Generative Adversarial Networks, the machine learning frameworks reached new heights of performance. GANs also majorly contributed to the rise of Generative AI. The idea behind the architecture of GANs has two RNNs training each other to learn from their errors, which is similar but not identical to the Transformer architecture that ChatGPT follows. (Figure 1 below shows the evolution of neural networks.)

Generative AI tools, predominantly text-based large language models like ChatGPT, BERT, etc., are built on transformer architecture.

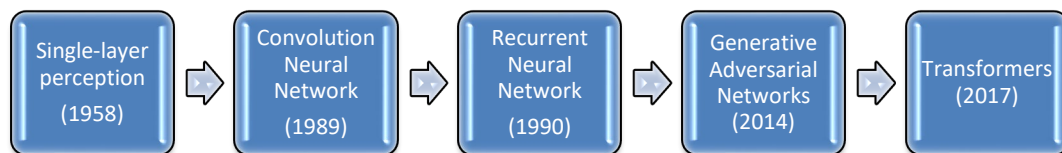


Figure 2.2: Evolution of Neural Networks

2.3 Recurrent Neural Networks

The origin of transformer architecture is based on Recurrent Neural Networks. Recurrent Neural Networks are feed-forward neural networks, which means the connections between neurons in the network do not form a cycle, which allows them to process sequential input, i.e., processing input data one after the other. Depending on the information and how it is fed to the RNNs, they are classified into three main categories: Vector-to-Sequence Models, Sequence-to-Vector Models, and Sequence-to-Sequence Models.

Vector-to-Sequence Model:

These neural nets have the input as a fixed-size vector, and the output is a sequence of any length. For Example, in the case of image captioning, the input is a vector representation of an image, and the output would be a sentence that describes the image.



Figure 2.3: Example of image captioning vector-to-sequence model

Sequence-to-Vector Model:

These neural networks take in a sequence and spit out a fixed-length vector. For Example, in sentiment analysis, a movie review is the input. The output is a vector indicating how good or bad the movie was according to this person, i.e., the probability of how good and bad a review can be is represented through a one-dimensional vector.



Figure 2.4: Example of movie review sentiment analysis sequence-to-vector model

Sequence-to-Sequence Model:

These neural networks take in a sequence and output another sequence. For Example, in language translation, the input could be a sentence in Spanish, and the output is its

translation in English.

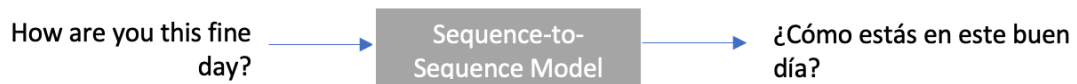


Figure 2.5: Example of language translation English to Spanish sequence-to-sequence model

One of the most popular applications that use this type of sequence-to-sequence model is by Google to build google translate for translating from one language to another.

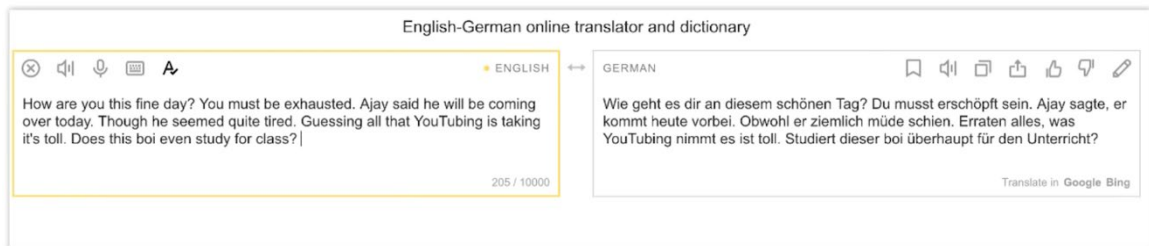


Figure 2.6: Visual of Google Translate application where language translation is in play

2.4 Neural Machine Translation with alignment:

Let's consider the case of language translation from English to French. The popular model as part of neural machine translation is the Encoder-Decoder model. The Encoder is an RNN neural network where a variable-length source sentence is encoded into a fixed-length vector. The Decoder is an RNN neural network where the encoded vector and the variable-length target sentence are fed as input.

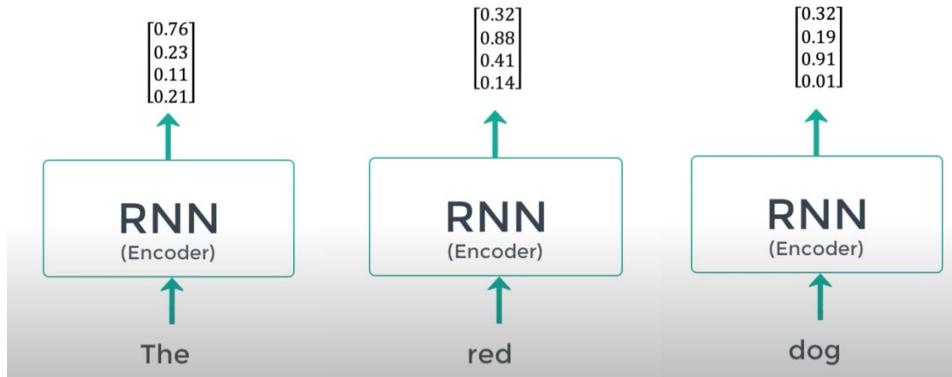


Figure 2.7: Visual representation of input passing in RNN Encoder [8]

Figure 2.5 above shows an example of RNN Encoder working, where the input words are fed one after the other, and the words are converted into fixed-length one-dimensional vectors one after the other. The RNN Encoder uses a Sequence-to-Vector model, whereas the RNN Decoder uses a Vector-to-Sequence model. Consider the Example of language translation from English to French using the RNN Encoder-Decoder model. The figure below shows a high-level representation of this task.

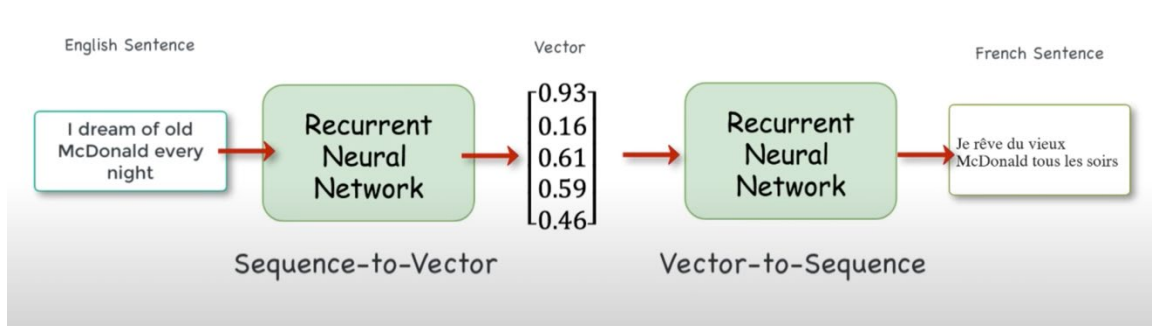


Figure 2.8: Visual representation of RNN Encoder-Decoder Model

Although the RNNs encoder-decoder model shows promising results, RNNs in combination with LSTM (Long Short-Term Memory) perform better at the same level as the conventional phrase-based translation models. Long Short-Term Memory (LSTM) Networks were introduced in 1991, which retain memory to deal with longer sequences better. [5] This paper describes how the basic encoder-decoder model can be further

improved by aligning and translating. This forms the basis of the attention mechanism used in the transformer model. Two modifications are done to the basic encoder-decoder model; one uses a bidirectional RNN encoder instead of a unidirectional Encoder; this allows the annotation to summarize the preceding words and the following words of the input sequence that's fed to the model. This sequence of annotations will then be fed into the decoder and alignment models to compute the context vector further. The alignment model is parameterized, a feed-forward neural network jointly trained with all the components. The alignment was not an active variable focused on the traditional machine translation model. Considering the alignment as a neural network model allows backpropagation with the help of a gradient descent mechanism to train itself and the whole translation model jointly. This sort of bidirectional encoder model is used in the architecture of a large language model used by Google called BERT. This further gave rise to the transformer model, where several of these encoder-decoder models are combined to form a large model. The transformer model architecture is described in detail in the next section.

2.5 Transformer model

The Transformer model introduced in 2017, employ Encoder and Decoder architecture much like RNNs; the difference is that the input sequence can be passed simultaneously and not sequentially. Consider the case of translating a sequence from English to French with an RNN encoder; English sentences are passed one word after the other, i.e., the current word's hidden states depend on the previous word's hidden states, as shown in Figure 2.5. Unlike RNN Encoder, input to Transformer Encoder has all the words of the sentence passed at once, as shown below in Figure 2.7.

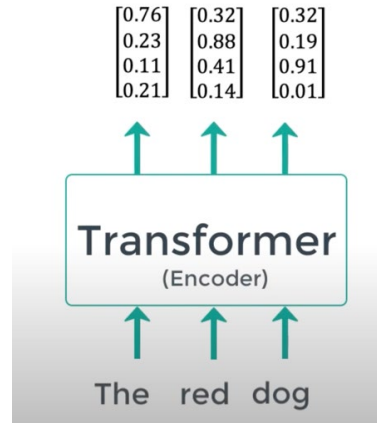


Figure 2.9: High-level Transformer Encoder working [8]

After the input sentences are passed, the word embeddings are generated in succession. In NLP, word embedding is when a word is mapped to a vector using similar words and meanings in the embedding space.



Figure 2.10: Example for input word embedding [8]

In Figure 2.8, the input word ‘dog’ is mapped to an embedded space where all the similar meaning or context words are placed together so that the distance between them is minimal. These distances from the input word to its neighboring words are then captured into a word embedding vector. But the same word in different sentences can have different meanings. Suppose our input sentence is as shown in Figure 2.9. The context of ‘dog’ in the first is a lot different than that of the ‘dog’ in the second sentence. This is where positional encoding comes into the picture. The positional encoding can be calculated in several ways. Still, the

method in [1] uses sine or cosine functions of varying frequencies to generate the final vector or embedding with context. Now we pass this vector into the Encoder block.

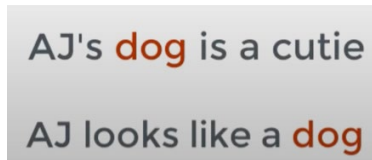


Figure 2.11: Example to explain positional encoding [8]

To review the process in the transformer Encoder architecture shown in Figure 2.10, all the words are passed in the sentence simultaneously, and determine the word embedding simultaneously. The word embeddings have position encoders. Position encoding is numerically tagging the words, in the form of vectors, according to their position in the sentence. After passing the input, i.e., English sentence to the input embedding and applying positional embedding, we get word vectors with context information. This is then passed into a multi-headed attention block.

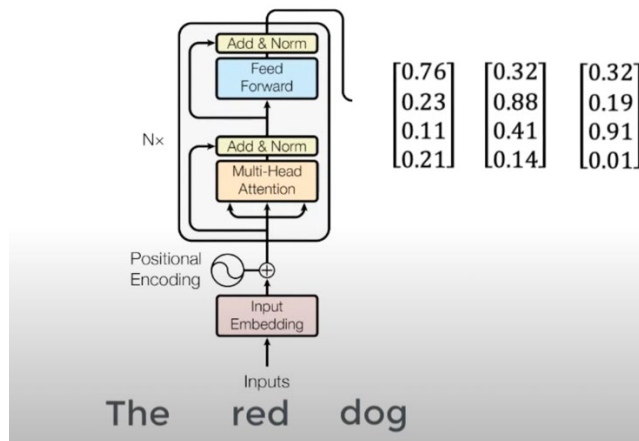


Figure 2.12: Transformer Encoder architecture [8]

[7] The attention block answers which part of the input should be focused on. Attention block represents contextual relationships between words in the sentence. The output of the attention vectors is passed into simple neural networks to be in a format

compatible with further Encoder or decoder blocks. The word embedding that is output at Encoder is passed into the decoder block that has three main components, two of which are similar to the encoder blocks. The self-attention block is where the mapping happens; in our Example of language translation from French to English, mapping the French words to corresponding English words occurs in the self-attention block.

The attention block generates vectors for each word in the French sentence related to every word in the same sentence. After passing through the multiple attention blocks, the output is attention vectors for every word in both English and French. This is further passed through a linear feed-forward layer and a probability distribution to predict the next word in the sentence.

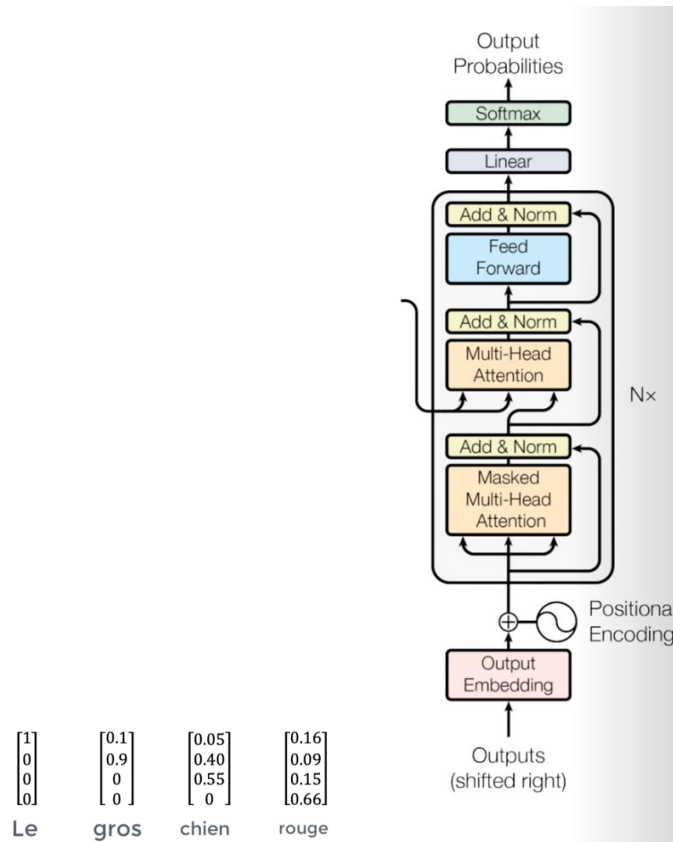


Figure 2.13: Transformer Decoder Architecture [8]

The Decoder block works like the Encoder block in the initial steps. In the training stage,

the Decoder is fed French sentences. Of course, since the computer wouldn't understand words, the input embedding vector with positional encoding is used to generate vectors. This is where the steps from Encoder and Decoder match.

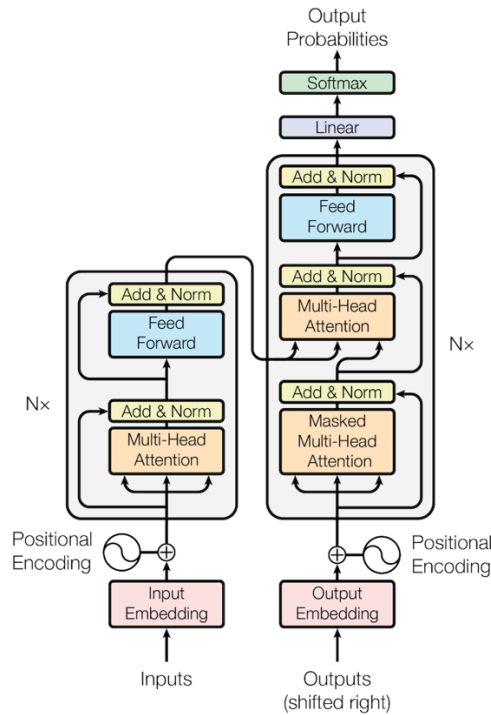


Figure 2.14: Transformer Encoder and Decoder Architecture [8]

The vector generated from French sentences is passed into an attention block along with the vectors generated out of the Encoder block. This is where the actual encoder and decoder mapping happen. The similarities of the vectors are calculated and ultimately give rise to the French sentences for the respective English sentences.

2.6 Reinforcement Learning by Human Feedback (RLHF)

ChatGPT is built on GPT and uses reinforcement learning from the Human feedback AI model, which will aid the active learning process, i.e., refined on the fly. The GPT models are Language models that employ the transformer architecture explained above. The ChatGPT is fine-tuned further using Reinforcement Learning by Human Feedback.

Reinforcement Learning is a method used to achieve a goal via rewards. The figure below shows the mechanism of reinforcement learning. The goal is for the agent to go to this end state or an unknown environment. To entice the agent to make certain moves, we use rewards. Rewards are scalar values given to each state; a high reward is given if the agent is close to the goal, and the rest have a low reward. Action is the actions the agent takes to achieve the goal. The policy is a sequence of actions an agent takes to accomplish the goal. [6].

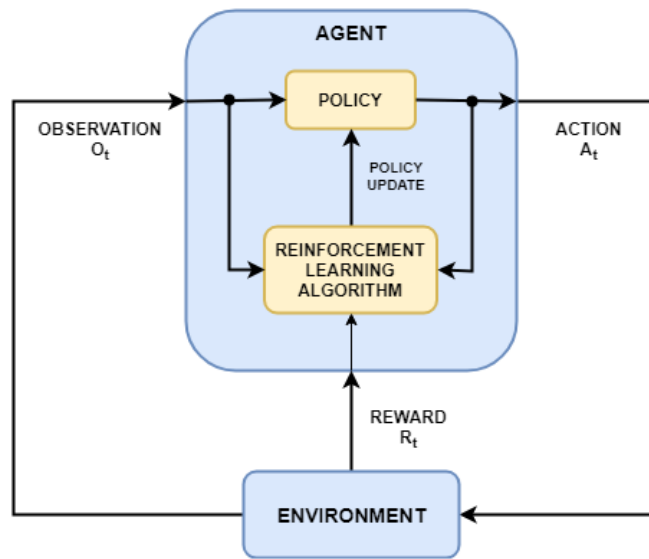


Figure 2.15: Reinforcement Learning [11]

Relating this to ChatGPT, the reward depends on the response given. If the answer given is good, it will get a high reward; if it's not, it will get a negative reward. The responses are provided by the labelers, i.e., users. In the context of ChatGPT, the action could be every word that the ChatGPT gives out, i.e., what word to generate next. So, the overall policy would be a sequence of actions taken. Each policy would have its reward, which can be compared to see which has achieved the highest reward and learn from it. This model is trained and fine-tuned through a couple of steps. In the supervised step, AI trainers converse with the AI model and provide responses to demonstrate the desired

output behavior. After the model is first trained, the sample of the model outputs was ranked best to worst by the AI trainers, and this course is used to train the AI model to calculate rewards for all the model outputs later, using this rewards model in the last step ChatGPT is fine-tuned using reinforcement algorithm. This reinforcement algorithm used by open AI is called Proximal Policy Optimization [7].

To investigate my research questions, I have considered the two courses from the UNCW Computer Science curriculum: the undergraduate level course CSC242 (Computer Organization) and a graduate level course MIS555 (Database Management Systems). I test the questions from these courses on the ChatGPT tool and analyze how accurate the ChatGPT responses are.

3.1 MIS 555 Database Management Systems:

The course curriculum of Database Management Systems at UNCW deals with topics on how to access and manipulate the data from database tables (structured data) using SQL programming. It also covers building relational database models, procedures, and common table expressions (CTEs).

3.2 CSC 242 Computer Organization:

The Computer Organization refers to the internal structure and implementation of a computer system. At UNCW, this course comprises topics like Machine-level data representation, Introduction to binary machine instructions on LC3, and assembly language instructions on LC3.

3.3 Types of questions in the courses:

The types of questions posed to ChatGPT are as follows:

- Theoretical questions that explore the logical concepts of the subject
- MySQL programming questions where the behavior of a few clauses that are specific to the tool are analyzed.
- Data representation: binary to decimal to hexadecimal conversion
- LC3 binary machine programming questions
- LC3 assembly language instructions

3.4 Crafting the prompts:

All the text that we enter in the ChatGPT as input is called a prompt. There are typically two types of prompts:

- Zero-shot learning prompts:

Zero-shot learning prompts are the input given to a language model to perform a task without specifying any examples. The prompts are like instruction, and the language model responds depending on how it is trained.

For Example, prompts like, “What is the capital of France?” although the language model may not explicitly be trained on such data, it might know this knowledge from tons of data that the model was trained on.

- Few shot learning prompts:

Few shot learning prompts are similar but slightly different in mentioning the examples or structure related to the task. This gives the language model to behave in a similar structure.

For Example, we give a prompt as below,

peppermint: menthe poivrée

desert cactus: cactus du désert

potato: pomme de terre

lipstick:

ChatGPT would respond as lipstick: rouge à lèvres. As the model understands, translating English words into French is a task.

Crafting clear and compelling prompts w.r.t the course content is crucial in providing context to the responses. The following steps have been followed in my project, which helped in effective, prompt response:

- Checked if the pre-existing relational database models exist for running SQL queries on them. (Northwind database)
- Asking theoretical questions on signed and unsigned binary representation first, then asking the data conversion questions.
- Asking for basic LC3 instructions definitions and explicitly following it up for binary machine instructions.

Ultimately, there isn't an optimal way to craft a prompt because it depends on the underlying model. Sometimes, slightly changing the word order or making minimal changes to the prompt might improve the responses in a not predictable way, as it depends on how the language model perceives the input.

3.5 Workflow of the Methodology:

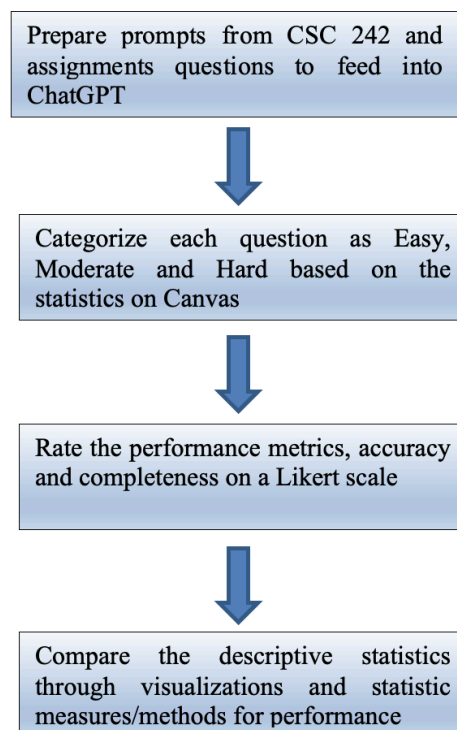


Figure 3.1: Workflow of Methodology

Performance measures: Accuracy and Completeness

Why did I consider accuracy and Completeness as performance measures in my project?

Accuracy and Completeness are two different metrics to evaluate data quality. Accuracy refers to the degree of correctness and validity of the data, i.e., which is free from errors and biases. Whereas Completeness refers to the comprehensiveness of the data, meaning how much accurate data has all the required components or facts.

Several methods are used to assess the performance of quantitative measures, such as accuracy, precision or recall, etc., be it classification or regression problems. But what metrics do we use to assess performance for text-based responses, especially the ones generated by ChatGPT, which cannot be out-and-out categorized as yes or no and right or wrong? Because the responses are often nuanced, i.e., it can often mislead the user into believing that the answer is accurate, it would not be a fair judgment if it were based on a binary classification approach. To properly analyze accuracy and Completeness, we follow a Likert scale approach as suggested in [9].

A Likert scale is a method used in rating survey questions, which is a selection of options ranging from one extreme possibility to another extreme and all the ones in between

In our case, we aim to have two kinds of Likert scales: accuracy and Completeness. By comparing the Actual Answer to the ChatGPT Answer, we match the options from 1-6 described below. We have found that the accuracy of the responses was believed to fall into the following 1 to 6 categories on the Likert Scale:

1	completely incorrect
2	more incorrect than correct

3	Approximately equal correct and incorrect
4	more correct than incorrect
5	nearly all correct
6	correct

Table 3.1: Accuracy Rating

By comparing the Actual Answer to the ChatGPT Answer, we match the options from 1-3, which are described below. The Completeness of the responses was believed to fall into the following 1 to 3 categories on the Likert Scale:

1	incomplete, addresses some aspects of the question, but significant parts are missing or incomplete
2	adequate, addresses all aspects of the question, and provides the minimum amount of information required to be considered complete
3	comprehensive, handles all aspects of the question, and provides additional information or context beyond what was expected

Table 3.2: Completeness Rating

CHAPTER 4: PRELIMINARY EXPERIMENTS

In this section, I aim to explore the behavior of ChatGPT to different types of assignment questions from the courses considered.

4.1 Experiment 1 (MIS 555):

Testing the tool by reframing the question.

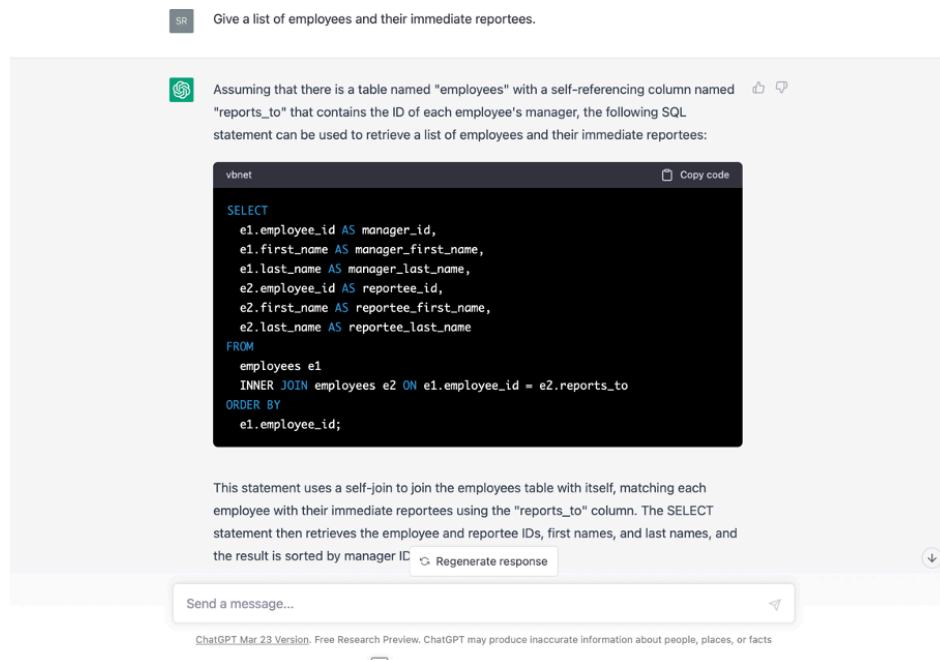


Figure 4.1: SQL Question 1

The above figure shows that ChatGPT has responded exceptionally well to an SQL query on the Northwind database. The ChatGPT has been trained on the existing sample database schemas, and the unique feature of ChatGPT allows it to write code blocks using the OpenAI Codex, a general-purpose programming model.

SR Give a list of employees, for each employee show every person above them.

Assuming that there is a table named "employees" with a self-referencing column named "reports_to" that contains the ID of each employee's manager, the following SQL statement can be used to retrieve a list of employees and every person above them:

```
SELECT
  e1.employee_id AS employee_id,
  e1.first_name AS employee_first_name,
  e1.last_name AS employee_last_name,
  e2.employee_id AS manager_id,
  e2.first_name AS manager_first_name,
  e2.last_name AS manager_last_name
FROM
  employees e1
  INNER JOIN employees e2 ON e1.reports_to = e2.employee_id
ORDER BY
  e1.employee_id,
  e2.employee_id;
```

This statement uses a self-join to join the employees table with itself, matching each employee with their manager using the "reports_to" column. The SELECT statement then retrieves the employee and manager last names, and the result is sorted by employee ID and manager ID.

Send a message...

ChatGPT Mar 23 Version. Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts

Figure 4.2: SQL Question 2

In Figure 4.2, the question posed to the ChatGPT was an extension to the previous question: instead of the immediate reporters of the employees, we ask the SQL query to show all the employees that each employee reports to; in other words, the hierarchy of the employee. But the above SQL query by ChatGPT still produces the same query as above. Sometimes, it might be how we word our questions that we don't get accurate answers. Let's try phrasing the prompt differently.

SR Give a list of employees, for each employee show all people above them in the hierarchy

Assuming that there is a table named "employees" with a self-referencing column named "reports_to" that contains the ID of each employee's manager, the following SQL statement can be used to retrieve a list of employees and all people above them in the hierarchy:

Figure 4.3: SQL Question 3

By reframing the previous question like above, we actually get the ChatGPT to give the desired response, i.e., the usage of recursion with common table expression, which is not explicitly mentioned.

```
sql Copy code  
  
WITH RECURSIVE employee_hierarchy AS (  
  SELECT  
    employee_id,  
    first_name,  
    last_name,  
    reports_to,  
    ARRAY[employee_id] AS hierarchy  
  FROM  
    employees  
  WHERE  
    reports_to IS NULL -- assuming the highest level manager doesn't have a  
  UNION ALL  
  SELECT  
    e.employee_id,  
    e.first_name,  
    e.last_name,  
    e.reports_to,  
    eh.hierarchy || e.employee_id  
  FROM  
    employees e  
    INNER JOIN employee_hierarchy eh ON e.reports_to = eh.employee_id  
)
```

Figure 4.4: SQL Question 3 (continued-part1)

```
SELECT  
  eh.employee_id,  
  eh.first_name,  
  eh.last_name,  
  e.first_name AS manager_first_name,  
  e.last_name AS manager_last_name  
FROM  
  employee_hierarchy eh  
  INNER JOIN employees e ON eh.reports_to = e.employee_id  
ORDER BY  
  eh.hierarchy DESC;
```

Regenerate response

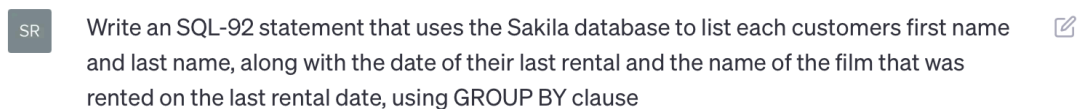
Figure 4.5: SQL Question 3 (continued-part2)

4.2 Experiment 2 (MIS 555):

Testing the tool on various versions of MySQL.

The most spoken-about functionality in SQL is GROUP BY clause. Because of its distinct behavior in various versions of MySQL, we test if ChatGPT would be able to differentiate the behavior of the GROUP BY clause. We test this over Sakila database which is an existing database consisting of video rental data. The challenge really comes with the mode ONLY_FULL_GROUP_BY, in earlier versions SQL (<5.7), when you try to add a non-aggregated column in the GROUP BY clause which is present in the SELECT clause, it gives an error. But, in the versions (>5.7) there is a mode ONLY_FULL_GROUP_BY which is disabled by default, when enabled it allows the above-described GROUP BY functionality.

In the below figure (Figure 4.7), we provide a scenario to the ChatGPT and ask to give us a query response in SQL-92 version, which is one of the earlier versions. Our anticipation is that it should only mention those columns in the GROUP BY clause that are aggregated and are present in the SELECT clause.

A screenshot of a ChatGPT prompt. On the left, there is a grey square icon with the letters 'SR' in white. To the right of the icon, the text reads: "Write an SQL-92 statement that uses the Sakila database to list each customers first name and last name, along with the date of their last rental and the name of the film that was rented on the last rental date, using GROUP BY clause". On the far right of the prompt, there is a small blue icon of a document with a pencil, indicating an edit function.

SR Write an SQL-92 statement that uses the Sakila database to list each customers first name and last name, along with the date of their last rental and the name of the film that was rented on the last rental date, using GROUP BY clause

Figure 4.6: SQL Question 4

But the response below (Figure 4.7) shows the SQL code resembling the behavior of version greater than 5.7. This shows that although the ChatGPT works great for most of the SQL related questions, it does have little bugs that are difficult to recognize and definitely require human surveillance.



Figure 4.7: SQL Question 4 (contd.)

4.3 Experiment 3 (MIS555):

Testing the tool with conceptual questions in DBMS.

ChatGPT's response to some of the conceptual questions is not what one expects but is not entirely incorrect. One cannot rule it out as right or wrong, as the response was very nuanced. To get the exact answer we anticipate, we have to modify the question by giving

them some context we are looking for. In the below Example, the expected solution is horizontal partitioning, which we do not get from ChatGPT immediately.

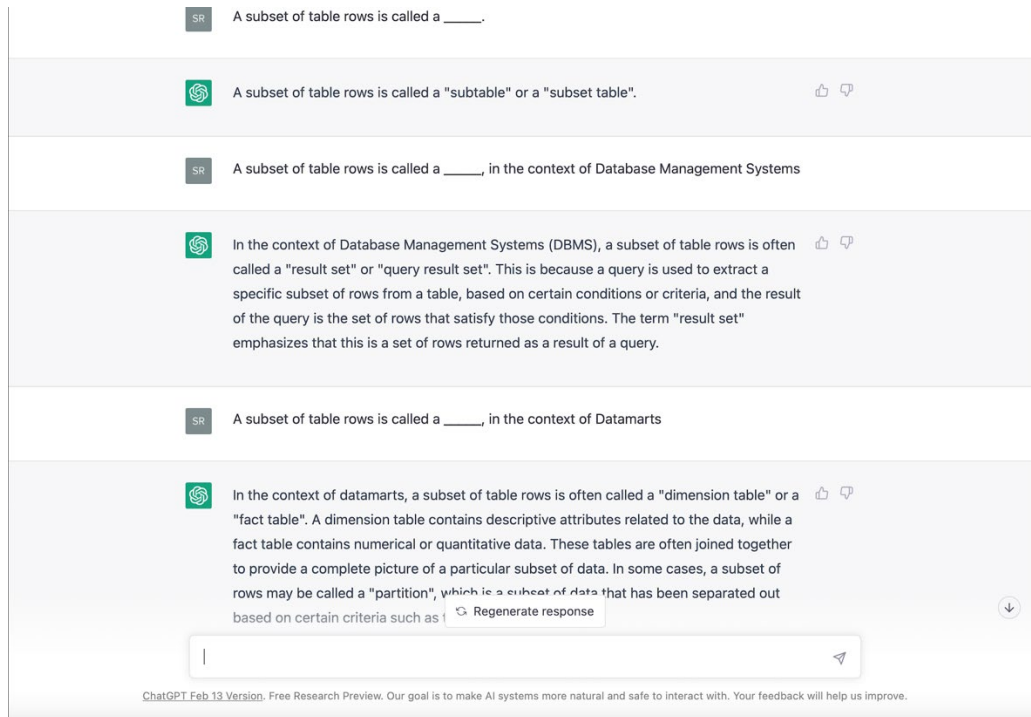


Figure 4.8: SQL Question 5

Still, when I give it some context, from my knowledge of encountering 'partitioning' in the context of DataMart, the response gets closer to what is expected but not quite. When asked for a definition of horizontal partitioning, we got the answer we hoped to see.

SR horizontal partition in database is called?



Horizontal partitioning in a database is also known as "sharding" or "shared-nothing partitioning". It is a technique in which a large table is divided into smaller tables based on the rows, so that each smaller table contains a subset of rows from the original table. This is typically done for performance or scalability reasons, as it allows the data to be distributed across multiple servers or nodes, each of which is responsible for a portion of the data. By partitioning the data horizontally, queries can be run in parallel on different subsets of the data, which can improve query performance.



Regenerate response

ChatGPT Feb 13 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

Figure 4.9: SQL Question 5 (continued)

4.4 Experiment 1 (CSC242):

Testing the tool with different versions of the model.



A higher-level programming language cannot instruct a computer to compute more than a lower-level programming language?



No, a higher-level programming language cannot compute more than a lower-level programming language. The capability of a programming language to perform computations is not determined by its level of abstraction.



In general, higher-level programming languages are designed to make it easier for the programmer to express their ideas, by providing abstractions that hide some of the details of the underlying hardware. These abstractions allow the programmer to write code that is more concise, readable, and maintainable, but they don't change the underlying capabilities of the computer.

Ultimately, all programs written in any programming language must be translated into machine code, which is a series of instructions that the computer can execute. The machine code generated from a higher-level programming language will have the same computational capability as the machine code generated from a lower-level programming language, since the underlying hardware is unchanged.

In other words, the choice of a higher-level or a lower-level programming language primarily affects the productivity and experience of the programmer, rather than the computational capability of the computer.



ChatGPT Jan 30 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

Figure 4.10: ChatGPT Jan 30 version

ChatGPT is still a preview in progress, and it is actively learning from human feedback. Here's an example where the ChatGPT version Jan30 has a different answer than the ChatGPT version Feb13. The question was tested on both versions, and although the earlier (Jan30) version response was incorrect, it seems to have been corrected in the Feb13 version, as shown below.

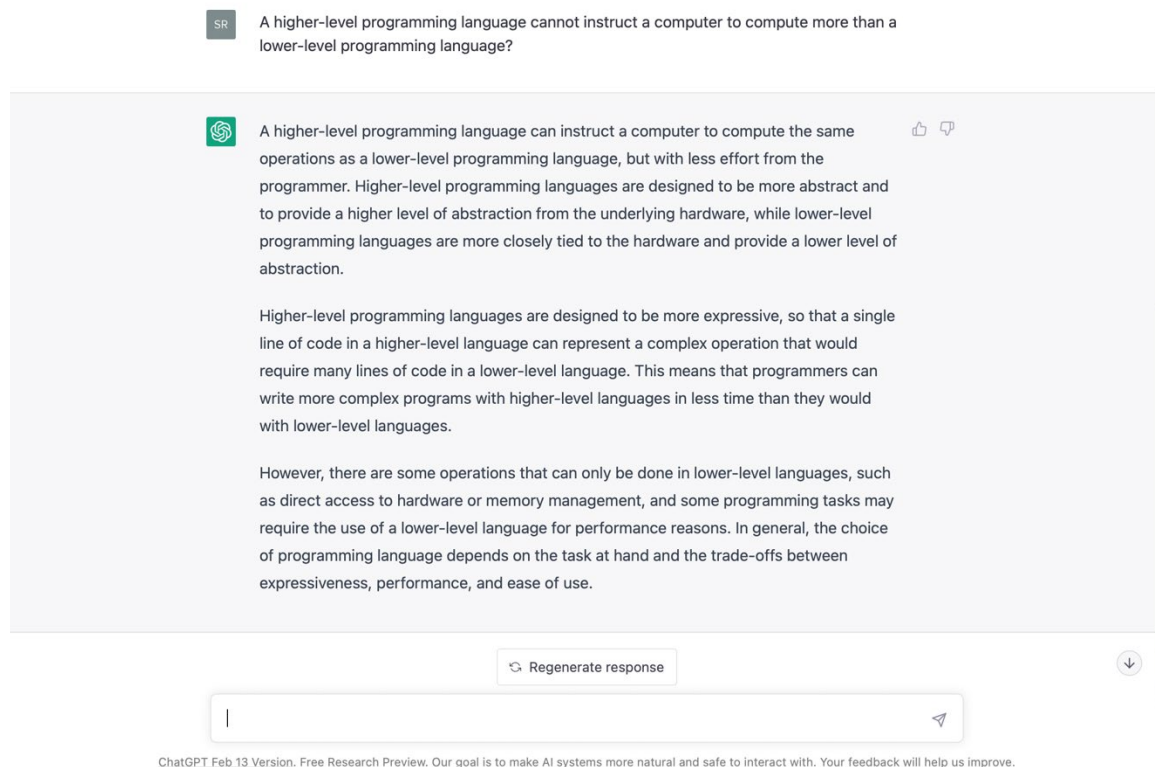
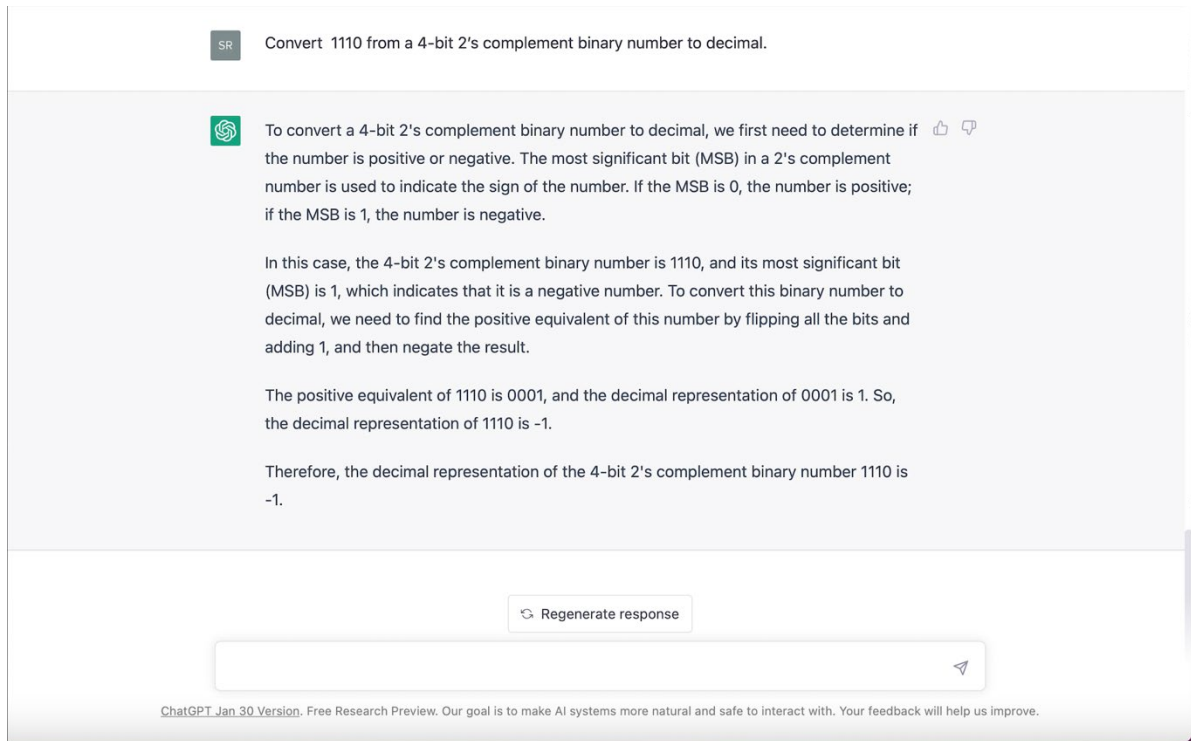


Figure 4.11: ChatGPT Feb 13 version

4.5 Experiment 2 (CSC242):

In a few of the questions answered by ChatGPT, especially when coding questions in LC3 binary instructions, the responses seem to approach the problem in the right direction conceptually. Still, while calculating the steps, it was incorrect.

In the Example below, if the MSB of a binary number is 1 then the number is negative, so we need to flip all the digits, add 1 to the prior result, and then convert the corresponding



The screenshot shows a chat interface with a question and an answer. The question is: "Convert 1110 from a 4-bit 2's complement binary number to decimal." The answer explains the process of determining the sign based on the MSB, finding the positive equivalent by flipping bits and adding 1, and then negating the result. It concludes that the decimal representation of 1110 is -1.

SR Convert 1110 from a 4-bit 2's complement binary number to decimal.

To convert a 4-bit 2's complement binary number to decimal, we first need to determine if the number is positive or negative. The most significant bit (MSB) in a 2's complement number is used to indicate the sign of the number. If the MSB is 0, the number is positive; if the MSB is 1, the number is negative.

In this case, the 4-bit 2's complement binary number is 1110, and its most significant bit (MSB) is 1, which indicates that it is a negative number. To convert this binary number to decimal, we need to find the positive equivalent of this number by flipping all the bits and adding 1, and then negate the result.

The positive equivalent of 1110 is 0001, and the decimal representation of 0001 is 1. So, the decimal representation of 1110 is -1.

Therefore, the decimal representation of the 4-bit 2's complement binary number 1110 is -1.

Regenerate response

ChatGPT Jan 30 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

Figure 4.12: CSC242 Question 1

binary number to decimal. But the ChatGPT mentions the process right but fails to add 1 to compute the 2's complement of the given binary number, resulting in an incorrect response. The correct answer must be -2.

I am testing on a few more binary to-decimal conversions. The answer for the below question is supposed to be 12755, but the calculation by ChatGPT was inconsistent. Even though the logic it follows is correct, the math is not valid.

SR

Convert 0011 0001 1101 0011 from 2's complement binary numbers to decimal.



To convert a binary number in 2's complement representation to decimal, we follow these steps:  


1. If the leftmost bit is 1, then the number is negative. In this case, we invert all the bits and add 1 to get the magnitude of the number.
2. If the leftmost bit is 0, then the number is positive. In this case, we simply convert the binary number to decimal.

In the given binary number 0011 0001 1101 0011, the leftmost bit is 0, so the number is positive.

Converting this binary number to decimal, we have:

```
markdown Copy code  
0011 0001 1101 0011 = 0*2^15 + 0*2^14 + 1*2^13 + 1*2^12 + 0*2^11 + 0*2^10 +  
= 0 + 0 + 8192 + 4096 + 0 + 0 + 512 + 256 + 128 + 0 +  
= 13107
```

Therefore, the given binary number 0011 0001 1101 0011 in 2's complement representation is equal to 13107 in decimal.

 Regenerate response



[GPT Feb 13 Version](#). Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

Figure 4.13: CSC242 Question 2

4.6 Experiment 3 (CSC242):

Testing the performance on LC3 instructions.

ChatGPT responses for LC3 binary machine instructions seem to be unreliable. For reference, the below figure has the LC3 instruction syntax.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD ⁺	0001				DR			SR1			0	00		SR2		
ADD ⁺	0001				DR			SR1			1	imm5				
AND ⁺	0101				DR			SR1			0	00		SR2		
AND ⁺	0101				DR			SR1			1	imm5				
BR	0000				n	z	p	PCoffset9								
JMP	1100				000			BaseR			000000					
JSR	0100				1	PCoffset11										
JSRR	0100				0	00		BaseR			000000					
LD ⁺	0010				DR			PCoffset9								
LDI ⁺	1010				DR			i			PCoffset9					
LDR ⁺	0110				DR			BaseR			offset6					
LEA ⁺	1110				DR			PCoffset9								
NOT ⁺	1001				DR			SR			111111					
RET	1100				000			111			000000					
RTI	1000				0000000000											
ST	0011				SR			PCoffset9								
STI	1011				SR			PCoffset9								
STR	0111				SR			BaseR			offset6					
TRAP	1111				0000			trapvect8								
reserved	1101															

Figure A.2 Format of the entire LC-3 instruction set. Note: + indicates instructions that modify condition codes

Figure 4.14: LC3 instructions [10]

The prompt below asks the ChatGPT to fill in the blanks related to computing the PC offset, $(x3003) + \text{offset} = x3020$; the offset is $x3020 - x3003 = x001D$, which translates to the offset value in binary as 0 0001 1101. Not only is the computation incorrect, the opcode, i.e., the first four digits that give the information on the type of instruction, was identified incorrectly 1010 stands for LDI, not LDR, as shown below.

```

SR X3000 0101 001 001 1 00000 ; R1<-0
X3001 0101 010 010 1 00000 ; R2<-0
X3002 1010 001 _____ ; R1<- MEM[MEM[x3020]]
X3003 0001 001 001 1 11111 ; R1<-R1-1, TOP
X3004 0000 100 _____ ; BRn HALT
X3005 0001 010 010 1 00001 ; R2<-R2+1
X3006 0000 111 _____ ; BRnzp TOP
X3007 1111 0000 0010 0101 ; TRAP x25, HALT
X3008
...
X3020 0011 0000 1010 0000; MEM X30A0
...
X30A0 0000 0000 0000 0010 ; X=2

```



Given the provided code, we need to fill in the two blanks:



X3002 1010 001 _____ ; R1<- MEM[MEM[x3020]]

X3004 0000 100 _____ ; BRn HALT

Solution:

X3002 1010 001 110000000; R1<- MEM[MEM[x3020]]

X3004 0000 100 111111111; BRn HALT

Explanation:

[Regenerate response](#)

Send a message...



[ChatGPT Mar 23 Version](#). Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts

Figure 4.15: CSC242 Question 3

Explanation:

In the instruction at X3002, we need to load the value from the memory address pointed to by the address stored in memory location x3020 into register R1. We can use the LDR instruction for this purpose. The offset between the base register and the memory address is 0 in this case, so we can use the immediate mode. Therefore, the binary representation of the instruction would be:

```
css Copy code  
  
1010 001 001100000; R1 <- MEM[MEM[x3020]]
```

In the instruction at X3004, we need to branch to the label HALT if the result in R1 is negative. We can use the BRn instruction for this purpose. The offset between the current PC and the label HALT is -1 in this case, so the offset would be 11111111 in 2's complement representation. Therefore, the binary representation of the instruction would be:

```
yaml Copy code  
  
0000 100 11111111; BRn HALT
```

↻ Regenerate response

and a message...

ChatGPT Mar 23 Version. Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts

Figure 4.16: CSC242 Question 3 (continued)

4.7 Experiment 4(CSC242):

Here's another example of an LC3 instruction prompt, the question demands immediate ADD instruction instead of the ADD instruction with two source registers. Therefore, the expected instruction is 0001 011 010 1 00000. The instruction provided by the ChatGPT

assumes the destination register and source register are the same (010), whereas the destination register, as mentioned, is R3(011).

SR

Write a single LC-3 instruction to move the value from R2 into R3 that does not require the use of a mask. _____

NOTE: The instruction should have the format: xxxx yyyy xxxx yyyy



One LC-3 instruction to move the value from R2 into R3 that does not require the use of a mask is: 👍 🗨

bash

Copy code

```
ADD R3, R2, #0
```

This instruction performs an addition operation where the value of R2 is added to the immediate value 0 and the result is stored in R3. Since the immediate value is 0, this instruction effectively copies the value in R2 into R3 without changing it.

The instruction has the binary format of `0001 010 010 000000`, which matches the specified format of `xxxx yyyy xxxx yyyy`. The first four bits `xxxx` represent the opcode for the ADD instruction, and the second four bits `yyyy` represent the source and destination registers, where R2 is the source register and R3 is the destination register in this case.

Figure 4.17: CSC242 Question 4

This shows that when ChatGPT is asked straightforward questions related to the LC3 instruction topic, it does give an accurate response. Still, it doesn't do a good job when there is a requirement for the ChatGPT to compute and configure the type of instruction to be modeled for a given scenario.

CHAPTER 5: ANALYSIS AND RESULTS

In this chapter, we focus on developing performance metrics to measure the validity and correctness of the responses provided by the ChatGPT. We make use of the responses, which are descriptive, acquired by feeding the UNCW curriculum assignment questions as prompts into the ChatGPT tool. The dataset curated in an excel format has the following columns.

The metadata of the dataset is as follows:

Columns	Description
Difficulty Level	Values: Easy, Moderate, Hard
Question	Values: Assignment questions
Actual Answer	Values: True answer
ChatGPT Answer	Values: ChatGPT response
Accuracy (1-6): Rating	Values: range from 1-6
Completeness (1-3): Rating	Values: range from 1-3

Table 5.1: Metadata of the curated dataset

Difficulty level:

Question type analysis based on statistics on Canvas. The values of this column are Easy, Moderate and Hard depending on the statistics on what percentage of students have attempted the questions correctly. The questions are categorized as follows:

- Easy – 90 -100 %
- Moderate – 50 – 90 %
- Hard – $\leq 50\%$

Question:

The column consists of text that are assignment questions that are modified for context and fed into the ChatGPT tool.

Actual Answer:

This column consists of long sequences of text that represent the actual truth answers of each of the questions that can further be considered as a comparison measure to the responses provided by ChatGPT.

ChatGPT Answer:

This column consists of long sequences of text that are captured from the ChatGPT responses for each of the line items.

Accuracy (1-6) Rating:

Depending on the response the values of this column are in the range of 1 to 6

Completeness (1-3) Rating:

Depending on the response the values of this column are in the range of 1 to 3. The responses that were completely incorrect on the accuracy scale (score of 1) were not graded on Completeness.

5.1 How to Rate Accuracy and Completeness?

To show how I rated the questions with different levels of accuracy and Completeness, I make use simple generic examples to explain.

Accuracy:

Consider the example statements given below:

- “The capital of France is Rome.”

This statement deserves a rating of 1, i.e., completely incorrect, because the capital of France is Paris.

- “Photosynthesis is the process of releasing carbon dioxide into the atmosphere.”

This statement is rated 2, i.e., more incorrect than correct, because at least it has few components of the correct answer, Photosynthesis is actually the process by which plants convert carbon dioxide and water into glucose and oxygen.

- “The average human body temperature is 37 degrees Celsius.”

This statement is rated 3, i.e., approximately equal correct and incorrect, because the average human body temperature is around 37 not exactly 37, it is different for different bodies.

- “The primary colors are red, blue, and green.”

This statement is rated 4, i.e., more correct than incorrect, it is true if we assume that these are the primary colors for additive mixing colors, as for subtractive mixing colors the primary colors are different.

- “Water boils at 100 degrees Celsius.”

This statement is rated 5, i.e., nearly all correct, but at different atmospheric pressure the water boils at different temperatures.

- “Mount Everest is the tallest mountain in the world.”

This statement is rated 6, i.e., correct as it is completely true.

Completeness:

Consider the example statements given below:

- 1 – Incomplete:

Question: “What are the main causes of climate change?”

Response: “The main causes of climate change are greenhouse gas emissions from

human activities like burning fossil fuels, deforestation, and industrial processes.”

Explanation: The response provides some aspects of the main causes of climate change but lacks additional information and context. It could be more comprehensive by including natural causes of climate change, historical trends, and the impact of different greenhouse gases.

- 2 – Adequate:

Question: “Explain the process of photosynthesis.”

Response: “Photosynthesis is the process by which plants convert sunlight into energy. Chlorophyll in plant cells absorbs sunlight, and carbon dioxide from the air is taken in through tiny pores called stomata. Through a series of chemical reactions, the plant combines carbon dioxide and water to produce glucose and oxygen.”

Explanation: The response covers all the essential aspects of photosynthesis, but it doesn’t go beyond the minimum requirements. Additional information on the importance of photosynthesis for the ecosystem or how it supports life on Earth could enhance the response.

- 3 – Comprehensive:

Question: “What are the major factors influencing the decline of bee populations, and what are the implications for biodiversity?”

Response: “The decline of bee populations is influenced by several factors. One significant factor is the use of neonicotinoid pesticides, which have been shown to be harmful to bees and other pollinators. Additionally, habitat loss due to urbanization and agricultural expansion has reduced the availability of suitable nesting sites and food sources for bees. Climate change is also playing a role, as it

disrupts the timing of flowering and affects the availability of nectar and pollen. The implications of declining bee populations for biodiversity are substantial. Bees are crucial pollinators for many plant species, including a significant portion of the world's food crops. Without bees, the reproduction of these plants is compromised, leading to decreased crop yields and potential food shortages. Furthermore, many animal species rely on the fruits and seeds produced by these plants as part of their diet. As a result, declining bee populations can trigger cascading effects throughout ecosystems, impacting both flora and fauna.”

Explanation: The response is comprehensive, covering all aspects of the question with additional information and context. It not only identifies the major factors contributing to the decline of bee populations but also explains the broader implications of this decline for biodiversity and the interconnectedness of ecosystems.

5.2 Number of questions considered:

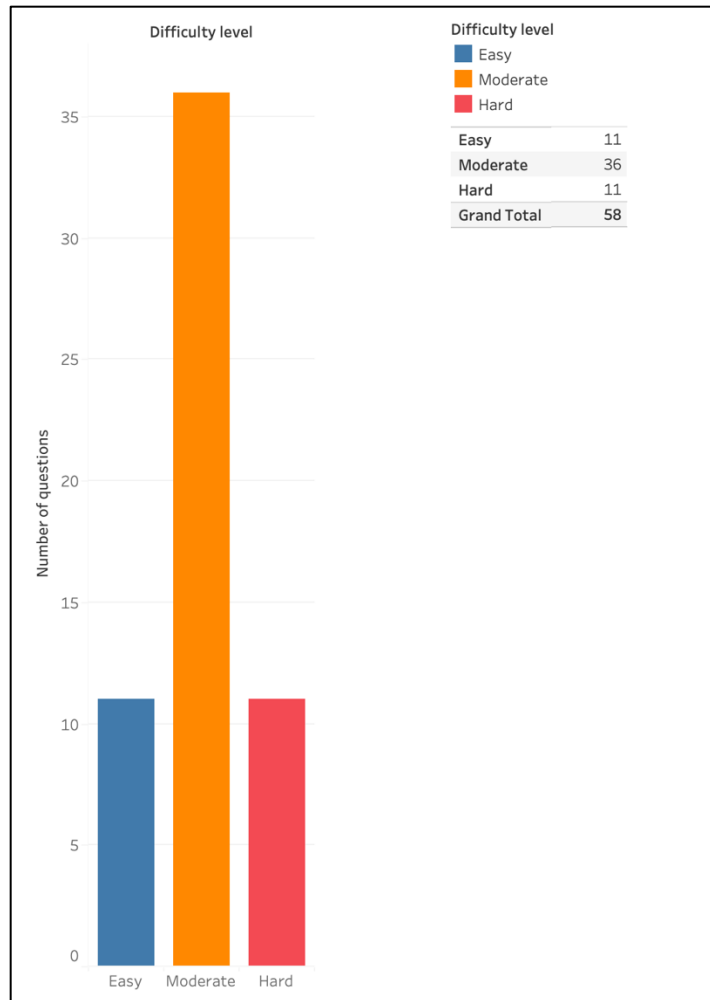


Figure 5.1: Count of the questions based on difficulty type

The above visualization shows a simple depiction of the number of questions considered for our case study. We see that total number of questions in the dataset is 58 out of which number of questions based on the difficulty level is as follows:

- Easy – 11
- Hard – 11
- Moderate – 36

5.2 Statistics of our dataset:

	Accuracy(1-6)	Count of Accuracy	%Accuracy
0	1	0	0
1	2	13	22.41
2	3	6	10.34
3	4	4	6.9
4	5	4	6.9
5	6	31	53.45

Figure 5.2: Percentage Accuracy calculation. Accuracy (1 – 6) (1- incorrect, 2 – more incorrect than correct, 3- approximately equal correct and incorrect, 4-more correct than incorrect, 5-nearly all correct, 6-fully correct)

	Completeness(1-3)	Count of Completeness	%Completeness
0	1	10	17.24
1	2	7	12.07
2	3	41	70.69

Figure 5.3: Percentage Completeness calculation. Completeness (1 – 3) (1- inadequate, 2 – adequate minimum requirements covered, 3- comprehensive)

```

*****ACCURACY*****
Median Accuracy: 6.0
Mean Accuracy: 4.59
Mode Accuracy: 6
SD Accuracy: 1.71
IQR Accuracy: 3.0
*****COMPLETENESS*****
Median Completeness: 3.0
Mean Completeness: 2.53
Mode Completeness: 3
SD Completeness: 0.78
IQR Completeness: 1.0

```

Figure 5.4: Descriptive statistics for the whole data

	Easy		Moderate		Hard	
	Accuracy	Completeness	Accuracy	Completeness	Accuracy	Completeness
Median	6.0	3.0	6.0	3.0	3.0	2.0
Mean	4.82	2.73	4.92	2.58	3.27	2.18
Mode	6	3	6	3	2	3
SD	1.6	0.47	1.66	0.81	1.42	0.87
IQR	2.0	0.5	2.25	0.0	2.0	1.5

Figure 5.5: Descriptive statistics based on the difficulty type

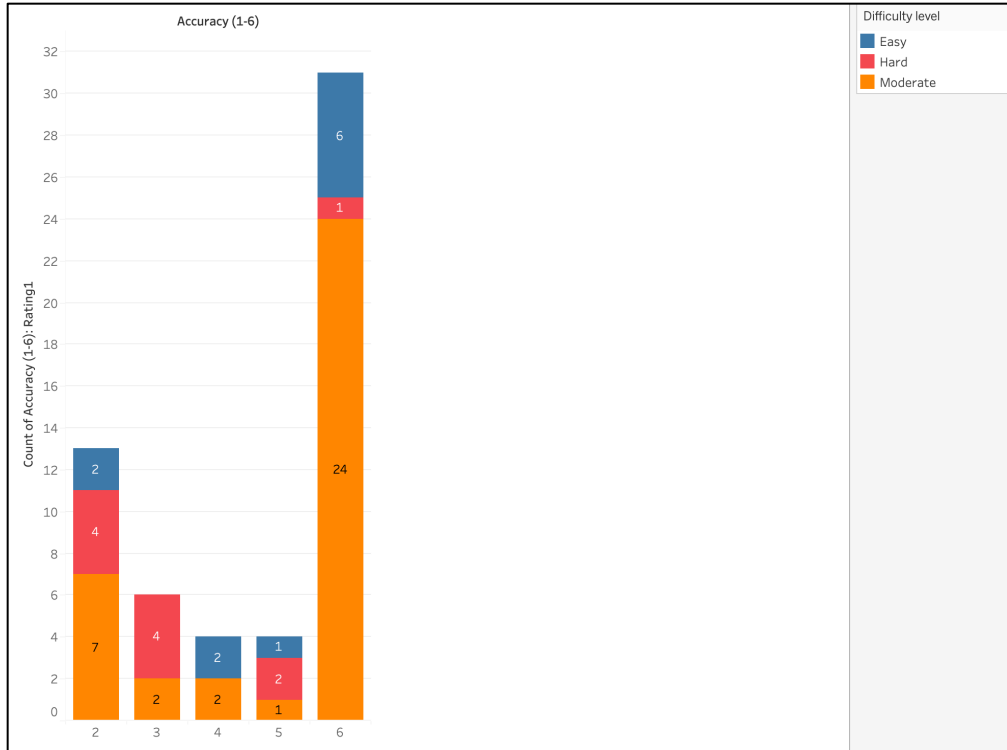


Figure 5.6: Stacked Accuracy plot

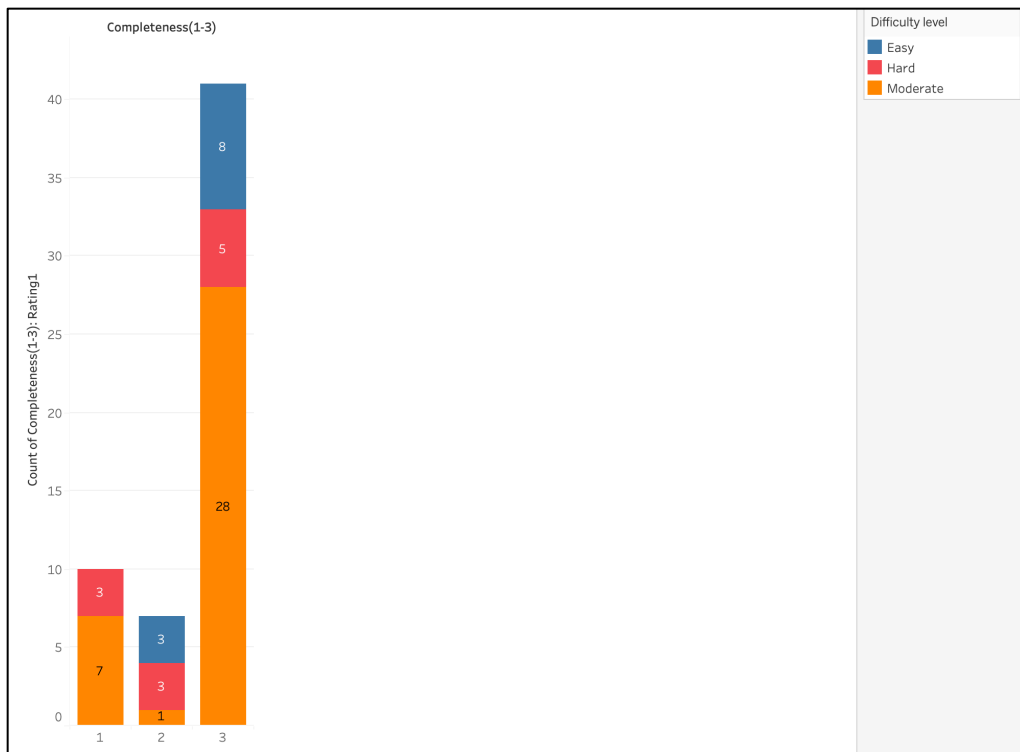


Figure 5.7: Stacked Completeness plot

Inference:

Median: is a measure that gives a view on the distribution of the data.

Mean: is a measure of the average distribution of the data.

Mode: is the measure that tells us the most frequently occurring value.

SD (Standard Deviation): is a measure that tells us how spread out the data is.

IQR (Inter Quartile Range): is a measure that tells us the distribution of the first 50% of the data

Among 58 ChatGPT-generated answers, referring to Figure 5.2, Figure 5.3, and Figure 5.4, we can see that the median accuracy score was 6 (mean 4.59, mode 6, SD 1.71), and the median completeness score was 3 (mean 2.53, mode 3, SD 0.78). 53.45% (n = 31) were scored at the highest level of accuracy (accuracy score of 6), and 6.9% (n=4) were scored as nearly all correct (accuracy score of 5). Conversely, 22.42% (n=13) of answers were scored as mostly incorrect (accuracy score of 2). Additionally, the Completeness of answers was evaluated, with 70.69% (n=41) scored as comprehensive, 12.07% (n = 7) as adequate, and 17.24% (n = 10) as incomplete. The inaccurate answers, receiving accuracy scores of 2 (n=13), were most commonly in response to hard and moderate questions. From Figure 5.5, we can see that only one response of difficulty level hard is rated with the highest accuracy. We can infer that ChatGPT couldn't provide clear and unambiguous response to questions that the students also found hard.

5.3 Correlation between accuracy scores of difficulty levels:

There are several ways to find out if there's relation between variables that are non-uniformly distributed. We explore this using Kruskal-Wallis Test and Mann-Whitney U Test. The two approaches give the statistical relationship between variables that were ranked. Kruskal-Wallis Test shows if there exists correlation between two or more

variables, but it doesn't tell which two variables have a correlation. To know what variables are correlated, we use Mann-Whitney U Test. We have considered the significance (alpha) = 0.05 as a measure to decide which null hypothesis is rejected. In statistics, the null hypothesis is the claim that no relationship exists between two variables, given the variables are not uniformly distributed.

Kruskal-Wallis Test:

By importing python scipy.stats package, and using the method kruskal to calculate the p-value as shown in the below figure. Accuracy scores between Easy, Moderate, and Hard levels are correlated across all questions as the p-value < 0.05; it satisfies the condition that the p-value need to be less than that of the alpha value.

```
1 # Kruskal-Wallis Test
2 from scipy import stats
3
4 # Defining data groups
5 data_group1 = easy_df['Accuracy (1-6): Rating1']
6 data_group2 = moderate_df['Accuracy (1-6): Rating1']
7 data_group3 = hard_df['Accuracy (1-6): Rating1']
8
9 # Conduct the Kruskal-Wallis Test
10 result = stats.kruskal(data_group1,data_group2,data_group3)
11 |
12 # Print the result
13 print(result)
```

```
KruskalResult(statistic=8.910127025879914, pvalue=0.011619581857389533)
```

Figure 5.8: Kruskal-Wallis Test

Mann-Whitney U Test:

Mann-Whitney Test, as described previously, is a test to find out if a correlation exists between two variables. The p-value is calculated by importing scipy.stats package in Python and using mannwhitneyu method, as shown below. The clause to know whether the correlation exists or not is the same as that of Kruskal's, which is the null hypothesis is rejected if p is greater than the assumed significance value of 0.05.

Easy and Moderate

```
1 # code for Mann-Whitney U test
2 from scipy.stats import mannwhitneyu
3 # Take batch 1 and batch 2 data as per above example
4 batch_1 = easy_df['Accuracy (1-6): Rating1']
5 batch_2 = moderate_df['Accuracy (1-6): Rating1']
6
7 # perform mann whitney test
8 stat, p_value = mannwhitneyu(batch_1, batch_2)
9 print('Statistics=%.2f, p=%.2f' % (stat, p_value))
10 # Level of significance
11 alpha = 0.05
12 # conclusion
13 if p_value < alpha:
14     print('Reject Null Hypothesis (Significant difference between two samples)')
15 else:
16     print('Do not Reject Null Hypothesis (No significant difference between two samples)')
```

Statistics=182.50, p=0.66

Do not Reject Null Hypothesis (No significant difference between two samples)

Figure 5.9: Mann-Whitney Test between Easy and Moderate

Moderate and Hard

```
1 from scipy.stats import mannwhitneyu
2 # Take batch 1 and batch 2 data as per above example
3 batch_1 = moderate_df['Accuracy (1-6): Rating1']
4 batch_2 = hard_df['Accuracy (1-6): Rating1']
5
6 # perform mann whitney test
7 stat, p_value = mannwhitneyu(batch_1, batch_2)
8 print('Statistics=%.2f, p=%.5f' % (stat, p_value))
9 # Level of significance
10 alpha = 0.05
11 # conclusion
12 if p_value < alpha:
13     print('Reject Null Hypothesis (Significant difference between two samples)')
14 else:
15     print('Do not Reject Null Hypothesis (No significant difference between two samples)')
```

Statistics=303.00, p=0.00405

Reject Null Hypothesis (Significant difference between two samples)

Figure 5.10: Mann-Whitney Test between Moderate and Hard

Hard and Easy

```
1 from scipy.stats import mannwhitneyu
2 # Take batch 1 and batch 2 data as per above example
3 batch_1 = hard_df['Accuracy (1-6): Rating1']
4 batch_2 = easy_df['Accuracy (1-6): Rating1']
5
6 # perform mann whitney test
7 stat, p_value = mannwhitneyu(batch_1, batch_2)
8 print('Statistics=%.2f, p=%.2f' % (stat, p_value))
9 # Level of significance
10 alpha = 0.05
11 # conclusion
12 if p_value < alpha:
13     print('Reject Null Hypothesis (Significant difference between two samples)')
14 else:
15     print('Do not Reject Null Hypothesis (No significant difference between two samples)')

```

Statistics=29.00, p=0.04
Reject Null Hypothesis (Significant difference between two samples)

Figure 5.11: Mann-Whitney Test between Easy and Hard

From the above figures, we can see that only the Easy and Moderate groups are modestly correlated, as they have a p-value greater than 0.05. This shows that the ChatGPT responses for the questions that are rated with the difficulty of Easy and Moderate are scored similarly. In our previous inference that ChatGPT performed fairly well with the questions of difficulty Easy and Moderate relative to the ones with difficulty level Hard, the Mann-Whitney U Test cements that same inference.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

This capstone project provides an early evidence base to demonstrate the potential of ChatGPT to questions from topics of Computer Organization or Architecture that are descriptive in nature. I was able to test to what extent ChatGPT can be used as a supporting tool for the students in the context of Computing Education as it has shown that more than half of the responses provided by ChatGPT are correct (53.45%) and comprehensive (70.69%). Even though it had an accuracy rate of 53.45 percent, we cannot entirely rely on the outcome of ChatGPT responses. It sure does provide a direction to understand a specific context or task, but assuming the output provided by ChatGPT is entirely correct is not fair as the responses are totally dependent on the underlying models, and the behavior of such language models is sometimes unpredictable.

As part of future work, testing this hypothesis on the larger dataset by adding more UNCW curriculum courses or, even better, testing it for topic-wise prompts in Computer Science, we can classify how strongly ChatGPT performs for certain areas and how we can rely on the tool. It would also be interesting to see how the recently released version GPT-4 would respond to the prompts and draw a comparison to the ChatGPT responses.

REFERENCES

Bibliography

- [1] N. S. N. P. J. U. L. J. A. N. G. L. K. I. P. Ashish Vaswani, "Attention Is All You Need," *arXivLabs*, 2017.
- [2] O. AI, "ChatGPT," Open AI , 30 November 2022. [Online]. Available: <https://openai.com/blog/chatgpt>.
- [3] O. AI, "Usage Policies," March 2023. [Online]. Available: <https://openai.com/policies/usage-policies>.
- [4] MathWorks, "Reinforcement Learning MathWorks," [Online]. Available: <https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html>.
- [5] K. C. Y. B. Dzmitry Bahdanau, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXivLabs*, 2014.
- [6] K. S. J. S. C. L. I. A. L. T. Shane Griffith, "Policy Shaping: Integrating Human Feedback with Reinforcement Learning," *NeurIPS*, 2013.
- [7] O. AI, "Proximal Policy Optimization," Open AI, 20 July 2017. [Online]. Available: <https://openai.com/research/openai-baselines-ppo>.
- [8] J. L., H. W., E. H., Y. S. Haifeng Wang, "Pre-Trained Language Models and Their Applications," *ScienceDirect* , 2022.
- [9] A. Halthor, "Transformers from scratch," 2023. [Online]. Available: <https://www.youtube.com/playlist?list=PLT19hO2Oobd97qfWC40gOSU8C0iu0m214>.
- [10] R. G. J. P. Douglas Johnson, "Assessing the Accuracy and Reliability of AI-Generated Medical Responses: An Evaluation of the Chat-GPT Model," *Research Square*, 2023.
- [11] S. P. Yale Patt, Introduction to Computing Systems: From Bits & Gates to C/C++ & Beyond.

APPENDIX

<https://github.com/savitharachuri/savy>

