

# Zero-shot-CoT on Logical and Positional Reasoning Tasks

Samyak Mehta  
sam012@ucsd.edu

June 24, 2024

## 1 Introduction

Initially, Large language models (LLMs) had demonstrated remarkable natural language understanding abilities, but their reasoning capabilities on multi-step arithmetic and logical problems were limited. These complex reasoning tasks require methodical step-by-step thinking to break problems down into smaller steps; a capability that has often eluded LLMs as model sizes increase.

Since then, a lot of work was done in this direction to improve the performance of LLM's on these tasks. Chain of thought prompting achieved significant success to improve the performance of earlier LLM models. One of these work was the Zero-shot-CoT (Kojima et al., 2023) which uses a simple prompt "Let's think step by step" that elicits multi-step reasoning from large language models across diverse tasks, without any task examples. It significantly improves zero-shot performance on arithmetic, symbolic, and logical reasoning benchmarks for e.g. increasing the accuracy on MultiArith (popular dataset comprising mathematical reasoning questions) from 17.7% to 78.7%.

Newer models of LLMs like gpt-3.5-turbo uses this multi-step thinking to better perform on such tasks. Despite this, there are some difficult logical reasoning and positional reasoning tasks on which the newer models still do not perform that well.

I will utilize the multiArith dataset, as employed by (Kojima et al., 2023) (The code for this paper is available at [https://github.com/kojima-takeshi188/zero\\_shot\\_cot](https://github.com/kojima-takeshi188/zero_shot_cot)), to demonstrate how GPT-3.5-turbo outperforms previous models used in their zero-shot-cot work. This improved performance is attributed to GPT-3.5-turbo's incorporation of chain of thought prompting. Additionally, I have created two datasets through web scraping: one containing approximately 50 instances of family relationship puzzles/questions and another with around 200

instances of seating arrangement questions, including circular, linear, square, and triangular arrangements.

In this study we find that, the GPT-3.5-turbo model finds it challenging to solve complex questions that require multi-hop reasoning. While GPT-3.5-turbo performs well on simple and structured questions from datasets like MultiArith (Roy and Roth, 2015), achieving an accuracy of 96.83%, it struggles with more complex and unstructured questions. Specifically, on datasets I created manually, such as the family relationship puzzle dataset and the seating arrangement dataset, it only achieves accuracies of 36.58% and 24.87%, respectively.

The analysis study paper shows several flaws in GPT-3.5-turbo's performance on complex logical and positional reasoning tasks:

1. The primary limitation is GPT-3.5-turbo's inability to handle multi-hop reasoning tasks effectively. While it performs well on simple, well-structured questions, it struggles to deliver on complex queries that contain multiple conditions or relationships in a single sentence.
2. Family relationship puzzles are another form of multi-layered relationships in which GPT-3.5-turbo can easily go wrong. It often overlooks key words (like "only") and sometimes it fails to correctly infer the final relationship, despite this step-by-step reasoning.
3. In particular, GPT-3.5-turbo struggles with relative directions like "left" and "right" in seating arrangements, which will depend on each person's perspective. It will also struggle when there are more than ten people or when arrangements span multiple rows or circles.

## 2 Dataset

The MultiArith dataset (Roy and Roth, 2015) is a collection of mathematical word problems designed to evaluate the arithmetic reasoning capabilities of language models. Each problem in the dataset typically involves a multi-step arithmetic operation, requiring the solver to interpret and perform a series of calculations based on the given textual description. (Kojima et al., 2023) showed how zero-shot-CoT improves performance on this dataset on earlier gpt-3 models, increasing the accuracy on MultiArith from 17.7% to 78.7% with large-scale InstructGPT model (text-davinci002). I will evaluate gpt-3.5-turbo on this dataset and it is expected to perform well since gpt-3.5-turbo does well on tasks that include chain of thought prompting.

Second dataset, which I manually webscraped, is Family Relationship Puzzle. This dataset consists of around 50 data points consisting of questions to deduce relationships between different family members based on given information, often depicted in the form of a family tree or through descriptive statements. We will see that gpt-3.5-turbo, although expected to perform well on logical reasoning questions, performs poorly here as this needs step-by-step multi-hop reasoning approach.

Third dataset, which I manually webscraped, is Seating Arrangement Dataset. This dataset consists of around 200 data points consisting of positional reasoning question involving arranging people in a particular way based on given positional and directional clues. We see that while gpt-3.5-turbo does well on linear arrangement, it does not perform well on circular arrangements. In such scenarios, LLMs often struggle due to the relative nature of directions like "right" and "left," which are contingent upon the orientation of the observer. Unlike linear cases where these directions remain consistent, they vary depending on each person's perspective at the circular table.

## 3 Analysis Approach

Firstly, I will evaluate GPT-3.5-turbo's performance on the MultiArith dataset (Roy and Roth, 2015), which is also used in the paper by (Kojima et al., 2023) that I am analyzing. Given that GPT-3.5-turbo incorporates chain-of-thought reasoning, it is expected to perform well on this dataset. I will determine its accuracy and analyze

cases where it still fails.

Secondly, I will assess GPT-3.5-turbo's performance on the Family Relationship Puzzle dataset, which I manually created through web scraping. This dataset contains complex questions and multi-hop reasoning, where GPT-3.5-turbo is anticipated to struggle.

Thirdly, I will examine GPT-3.5-turbo's performance on the Seating Arrangement dataset, also manually created by web scraping. Similar to the Family Relationship Puzzle dataset, this dataset features complex questions and multi-hop reasoning, where GPT-3.5-turbo is expected to encounter difficulties.

The process to prompt gpt-3.5-turbo is shown in Figure 1.

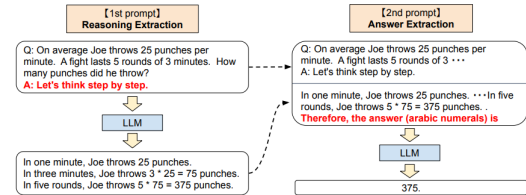


Figure 1: Pipeline for Prompting a done in (Kojima et al., 2023)

The analysis Approach for each dataset is listed below:

### 1. MultiArith Dataset:

**Method:** The study uses the same prompting technique as in Kojima et al. (2023), applying it to GPT-3.5-turbo.

**Interesting Aspect:** By using the same dataset and prompting technique as the original paper, the study provides a direct comparison of GPT-3.5-turbo's performance (96.83% accuracy) with earlier models like text-davinci002 (78.7% accuracy). This highlights the model's improved chain-of-thought capabilities.

### 2. Family Relationship Puzzle Dataset:

**Method:** I manually web-scraped around 50 complex family relationship questions and evaluated GPT-3.5-turbo using the same prompting technique.

**Interesting Aspect:** Despite the small dataset size, the analysis provides rich insights. I categorized the errors into four types, revealing nuanced issues like the

model’s struggle with multi-layered relationships and sensitivity to key words like ”only.”

### 3. Seating Arrangement Dataset:

Method: Another manually web-scraped dataset of about 200 questions, covering various seating arrangements (linear, circular, square, triangular).

The analysis breaks down performance by arrangement type 1, revealing that GPT-3.5-turbo struggles across all types, not just circular ones. This systematic breakdown helps pinpoint that the issue lies in complex, multi-constraint problems rather than just circular arrangements.

By categorizing these errors as different types (as done in Section 4), this qualitative analysis extends the model beyond direct accuracy metrics and hints at patterns to failures of this model, as in misinterpretation due to ambiguous wordings (MultiArith), struggles with multi-layered relationships (Family Puzzles), and failure to handle relative directions (Seating Arrangements).

This side-by-side comparison shows very clearly how prompt complexity directly affects GPT-3.5-turbo performance, which is very relevant to our proposed solutions.

Overall, methods are very simple (prompting, measuring accuracy, and analyzing errors), and systematic categorization of the errors, followed by the comparative analysis of simple vs. complex prompts, will bring deep insights, beyond mere surface metrics on understanding the nuanced limitations of GPT-3.5-turbo for guiding the development of the proposed pipelines for improvement.

## 4 Errors and their Categorization

### 4.1 MultiArith Dataset

On running the prompting technique as shown in Figure 1 using GPT-3.5-turbo on MultiArith (Roy and Roth, 2015) dataset, I get an accuracy of 96.83%. Zero shot CoT prompting on earlier models like text-davinci002 on MultiArith produced accuracy of 78.7%. This shows that gpt-3.5-turbo does better on logical reasoning tasks as compared to earlier models as it has better chain of thought capabilities as compared to earlier models. This dataset consists of around 600 examples.

GPT-3.5-turbo produced the wrong answer on just 19 instances. On analyzing the errors I find out that the errors are due to ambiguous text in the question and misleading wordings. I can categorize these errors into two types:

1. Type1: As shown in Figure 2 GPT-3.5-turbo misinterprets the question, but this is due to the question’s ambiguity. The question doesn’t specify whether the friends’ papers should be combined before calculating the points. The model assumes that Haley and her friend trade their papers separately, leading to the solution:  $\text{total-points} = \text{floor}(x/3) + \text{floor}(y/3)$ . However, the correct approach is to combine all papers first:  $\text{floor}((x+y)/3)$ . There are a total of 11 instances that roughly fall into this category of errors.
2. Type2: As shown in Figure 3 GPT-3.5-turbo misinterprets the question due to its phrasing. The question states, ”If each player left had 8 lives,” referring to the players still in the game. However, the model incorrectly interprets it as the players who left the game having 8 lives. There are roughly 6 instances of such type of errors where the model did not interpret the question correctly due to the way it was framed, although the rest of the logical reasoning was correct.

### 4.2 Family Relationship Puzzles Dataset

On running the prompting technique as shown in Figure 1 using GPT-3.5-turbo on this dataset, I get an accuracy of 36.58%. We see that GPT-3.5-turbo performs poorly on this dataset too, but better as compared to the other two datasets. This could be because the size of this dataset is small (around 50 samples).

But for analysis purposes, 50 data points is enough. Upon analyzing the dataset and the responses from GPT-3.5-turbo, I conclude that GPT-3.5-turbo provides correct answers for questions that are simply formed and well-structured, as illustrated in Figure 4. However, it does not perform well for questions that are complex and contain multiple constraints and conditions within a single sentence, as shown in Figure 5.

The errors can be categorized into four types:

1. GPT-3.5-turbo generally fails for those example where more than 3 dependencies and relationships are stated in a single sentence for

-----

Prompt: Q: Haley and her friends were recycling paper for their class. For every 3 pounds they recycled they earned one point. If Haley recycled 11 pounds and her friends recycled 16 pounds, how many points did they earn?

A: Let's think step by step.

Response: Step 1: Calculate how many points Haley earned for recycling 11 pounds of paper.  
 $11 \text{ pounds} / 3 \text{ pounds per point} = 3 \text{ points}$

Step 2: Calculate how many points Haley's friends earned for recycling 16 pounds of paper.  
 $16 \text{ pounds} / 3 \text{ pounds per point} = 5 \text{ points}$

Step 3: Add the points Haley earned to the points her friends earned.  
 $3 \text{ points} + 5 \text{ points} = 8 \text{ points}$

Therefore, Haley and her friends earned a total of 8 points for recycling paper.

Prediction: 8  
Ground Truth: 9

-----

Figure 2: Error Category Type1 on MultiArith Dataset

-----

Prompt: Q: There were 10 friends playing a video game online when 7 players quit. If each player left had 8 lives, how many lives did they have total?

A: Let's think step by step.

Response: First, we need to find out how many players left the game. Since there were originally 10 friends playing and 7 players quit, there are  $10 - 7 = 3$  players remaining.

Next, we need to calculate how many lives the players who left had in total. Since each player had 8 lives and 7 players left, the total number of lives they had was  $8 * 7 = 56$  lives.

Finally, we need to calculate how many lives the remaining players have. Since there are 3 players remaining and each player had 8 lives, the total number of lives the remaining players have is  $3 * 8 = 24$  lives.

Therefore, the total number of lives the players had in total is 56 (from the players who left) + 24 (from the remaining players) = 80 lives.

Prediction: 80  
Ground Truth: 24

-----

Figure 3: Error Category Type2 on MultiArith Dataset

-----

Prompt: Q: Rajan said, Pointing to Lalit in Photograph, 'His mother has only one grandchild whose mother is my sister.' How is Rajan related to Lalit? Answer Choices: (A) Brother (B) Father-In-Law (C) Brother-In-Law (D) Data Inadequate

A: Let's think step by step.

Response: First, we know that Lalit's mother has only one grandchild. This means that Lalit is the only child of his mother.

Next, we know that the mother of Lalit's child is Rajan's sister. This means that Rajan is the father of Lalit's child.

Therefore, Rajan is Lalit's brother-in-law.

So, the correct answer is (C) Brother-In-Law.

Prediction: C  
Ground Truth: C

-----

Figure 4: Simple and well structured Prompt and the response

-----

Prompt: Q: Richa said Introducing a boy to her mother, that he is the son of the daughter of the father of her uncle. How is Richa related to the boy? Answer Choices: (A) Brother-in-law (B) Brother (C) Father (D) Uncle (E) Son

A: Let's think step by step.

Response: First, let's break down the statement: "he is the son of the daughter of the father of her uncle."

This means that the boy is the son of the daughter of Richa's uncle's father. In other words, the boy is the son of Richa's cousin (daughter of her uncle).

Since the boy is the son of Richa's cousin, Richa would be the cousin's aunt. Therefore, Richa is related to the boy as his (C) Father.

Prediction: C  
Ground Truth: B

-----

Figure 5: Complex and unstructured Prompt and the response

e.g. "he is the son of the daughter of the father of her uncle".

- The model often struggles with multi-layered relationships where more than one intermediate relative needs to be identified. For exam-

ple: "Nilama said, Introducing Ramprakash, His brother's father is the only son of my grandfather." The model misidentified Nilama's relationship to Ramprakash, missing that Nilama must be Ramprakash's sister.

3. The model often overlooks the significance of words like "only" which changes the relational mapping significantly. For example: "Reena said, Pointing to a boy in the photograph, 'He is the only son of the only child of my grandfather.'" The model misinterpreted the extent of exclusivity marked by "only."
4. The model sometimes fails in correctly identifying the end relation after moving step by step from the starting point to the end relation. For example: "Abhishek said, Pointing to a girl, 'She is daughter of the only child of my father.' How is Abhishek related to that girl?" The model incorrectly concluded the answer as "Sister" instead of "Father."

### 4.3 Seating Arrangement Dataset

On running the prompting technique as shown in Figure 1 using GPT-3.5-turbo on This dataset, I get an accuracy of 24.87%.

This dataset consists of seating arrangement in a linear way or circular way and few example of square or triangular ways too. On breaking down the accuracies for each shape have the accuracies as shown in Table 1. We see that GPT-3.5-turbo performs almost equally poorly on all types of seating arrangement questions.

Shape	Total	Correct	Accuracy (%)
Straight line	95	24	25.26
Circle	82	20	24.39
Other shapes	20	5	20.00

Table 1: Accuracy for different seating arrangements

Upon analyzing the dataset and the responses from GPT-3.5-turbo, I conclude that GPT-3.5-turbo provides correct answers for questions that are simply formed and well-structured, as illustrated in Figure 6 However, it does not perform well for questions that are complex and contain multiple constraints and conditions within a single sentence, as shown in Figure 7.

The errors can be categorized into three types:

1. Generally, when the number of people to be arranged exceeds 10, GPT begins to struggle due to the increased context, which makes the sentences more complex.
2. Situations where the question requires arranging people in two rows or two circles become quite complicated, and GPT often makes mistakes in determining the correct row or circle for each person.
3. When arranging people in a circle, GPT struggles with the relative nature of directions like "right" and "left," which depend on the observer's orientation. Unlike linear arrangements where these directions remain consistent, they vary based on each person's perspective at the circular table.

## 5 Discussion

The errors observed in GPT-3.5-turbo's performance on complex logical and positional reasoning tasks can be attributed to several factors, drawing insights from the literature in Natural Language Processing (NLP).

1. Lack of Multi-hop Reasoning: As evidenced by the performance on the Family Relationship Puzzle and Seating Arrangement datasets, GPT-3.5-turbo struggles with tasks that require multi-hop reasoning. This is because these models primarily rely on pattern matching and lack a structured reasoning process.
2. The errors in the MultiArith dataset, particularly Type 1 errors, can be attributed to the inherent ambiguity in natural language. As pointed out by (Liu et al., 2021), language models can be sensitive to the way questions are phrased, leading to incorrect interpretations. This is a known challenge in NLP, where even slight changes in wording can significantly alter the meaning and confuse models.
3. The poor performance on circular seating arrangements with the challenges of spatial reasoning in language models. LLMs struggle in understanding relative directions like "left" and "right" in a circular context as it requires a form of perspective-taking that current language models lack. This is because these models are primarily trained on text, which doesn't inherently encode spatial relationships.
4. Scale of the Problem: As noted in the analysis, GPT-3.5-turbo's performance degrades



-----

Prompt: Q: O, P, Q, R, S, T and U are sitting in a straight line all facing north direction, but not necessarily in the same order. There is only one person sitting between T and Q. S sits between O and R. There are only two persons between S and U. T sits on the immediate left of O, who sits in the middle of the row. Which of the following persons is sitting between O and R? Answer Choices: (A) S (B) P (C) Q (D) U (E) None of these

A: Let's think step by step.

Response: 1. T sits on the immediate left of O, who sits in the middle of the row. This means that the order from left to right is T, O, and then the rest of the people.

2. There is only one person sitting between T and Q. Since T is on the immediate left of O, Q must be to the right of O. Therefore, the order is T, O, Q, and then the rest of the people.

3. S sits between O and R. Since O is in the middle of the row, S must be to the left of O and R must be to the right of O. Therefore, the order is T, S, O, R, Q, and then the rest of the people.

4. There are only two persons between S and U. Since S is to the left of O, U must be to the right of R. Therefore, the order is T, S, O, R, U, Q, and then the rest of the people.

Therefore, the person sitting between O and R is S. So the answer is (A) S.

Prediction: A

Ground Truth: A

Figure 6: Simple and well structured Prompt and the response

-----

Prompt: Q: Ten Friends Harish, Ratan, Manish, Arpit, Vivek, Daya, Rajan, Aman, Sachin and Rohit are seated in two rows facing each other. 5 of them are facing north and 5 are facing south but not necessarily in the same order. The arrangement is based on the following rules: Harish is seated 3rd to the right of Arpit. Ratan is to the immediate right of the person who is sitting in front of Arpit. Arpit and Vivek are the immediate neighbours of Sachin. Rajan is 4th to the left of the person who is in front of Aman. Aman is 2nd to the left of Rohit. Manish is facing the north and is seated to the left of Daya. Who among the following persons is/are sitting at the extreme end of the row? Answer Choices: (A) Vivek (B) Arpit (C) Ratan (D) Either Vivek or Arpit (E) Either Arpit or Rohit

A: Let's think step by step.

Response: 1. Harish is seated 3rd to the right of Arpit.

2. Ratan is to the immediate right of the person who is sitting in front of Arpit.

3. Arpit and Vivek are the immediate neighbours of Sachin.

4. Rajan is 4th to the left of the person who is in front of Aman.

5. Aman is 2nd to the left of Rohit.

6. Manish is facing north and is seated to the left of Daya.

From the given information, we can deduce the following:

- Arpit is in the middle of the row, with Harish to his right and Ratan to his left.
- Sachin is seated next to Arpit, so Sachin must be on one side of Arpit.
- Aman is 2nd to the left of Rohit, so Aman must be to the left of Rohit.
- Manish is facing north and is seated to the left of Daya, so Manish must be on the left side of the row facing north.

Based on the above deductions, the persons sitting at the extreme ends of the row are Vivek and Rohit. Therefore, the answer is (D) Either Vivek or Arpit.

Prediction: D

Ground Truth: C

Figure 7: Complex and unstructured Prompt and the response

as the number of entities (e.g., people in seating arrangements) increases. This is consistent with the concept of "semantic drift" discussed by (Gua et al., 2020). They argue that as the reasoning chain gets longer, language models are more likely to veer off course, introducing compounding errors.

5. The analysis reveals that GPT-3.5-turbo performs better with simple, well-structured questions. This aligns with the work of (Kojima et al., 2023), who introduced the "Let's think step by step" prompt. They argue that explicit instructions can guide language models towards more structured reasoning. However, as seen in this study, such prompts may not be sufficient for highly complex tasks.

## 6 The way forward

As we discussed in the previous Sections, we can see that the gpt-3.5-turbo does not perform well

on complex logical and positional reasoning tasks. The main reason for this is because gpt-3.5-turbo does not have multi-hop reasoning capabilities but such questions can perform better if multi-hop reasoning approach is used. I first start with proposing a framework to improve accuracy on the family relationship puzzle dataset and then emphasize on how this can be extended to the seating arrangement dataset too.

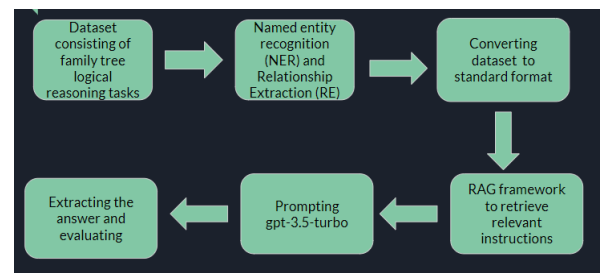


Figure 8: Proposed Pipeline for Family Relationship Dataset

My proposed pipeline for solving family tree logical reasoning tasks involves six key steps as shown in the Figure 8:

1. **Dataset Preparation:** This is the dataset consisting of complex family tree logical reasoning tasks.
2. **Use Named Entity Recognition (NER) to identify and extract relevant entities, including the names of people, and Relationship Extraction (RE) to determine the relationships between those entities that the dataset involves.** Many existing models can be applied in this step, including NER and RE models, POS tagging, and even entity recognition using tokenization in NLTK or a RoBERTa model that is finetuned for entity recognition or we can get this done by gpt too since gpt's are generally good at this. This will help to simplify the complex sentences; any dataset can include indirect and complex sentences, which can sometimes distract GPT-3.5-turbo. Questions having multiple conditions and implications merged into one statement often don't perform well while directly presented to GPT. Therefore, they need to be converted into simpler forms. It allows to convert the data into a standard format which greatly helps with the generalizability of the downstream framework.
3. **Standardization:** The dataset is then converted into a standard format, ensuring consistency and ease of processing.
4. **RAG Framework:** Implement a Retrieval-Augmented Generation (RAG) framework for retrieving instructions and context from a large corpus of data. These instructions could also be manually created as well as formatted, like, for instance, "If A is B's sister and B is C's brother, then A is C's sister." The obtained simplified sentences from the NER and RE framework are matched with the instruction set (using their word embeddings), one can extract (retrieve) the relevant instructions (one for each relation in the question). This way, the system will have a better potential for understanding and maneuvering complex relationships in the data.
5. **Prompting GPT-3.5-Turbo:** Next, prompt

GPT-3.5-turbo with the retrieved instructions and standardized data to generate answers.

6. Finally, Extract the answers provided by GPT-3.5-turbo and reformat it according to the format wanted. You can use some post-processing script to extract the answer from the response.

This proposed pipeline should definitely work better, as evidenced by examples where GPT-3.5-turbo struggles with complex, un-structured sentences but performs well with direct and structured sentences. By simplifying complex sentences through the NER and RE framework and using the RAG framework to retrieve relevant instructions, we can significantly enhance the model's ability to process and understand the data effectively.

Similar pipeline can be made for the Seating arrangement dataset. As we have seen that in GPT-3.5-turbo does well for linear arrangements but does not do well for circular or closed loop arrangements due to the relative right/left orientation depending on where the person is seated. So I have created a pipeline to address this issue as shown in Figure 9

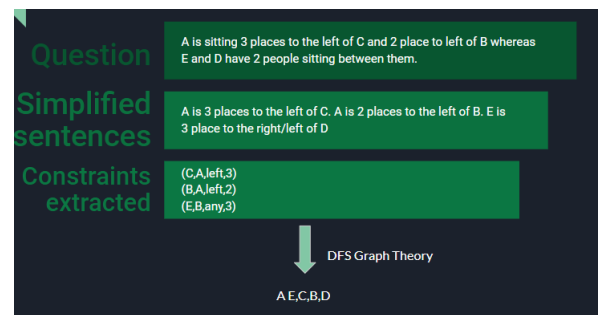


Figure 9: Proposed Pipeline for Seating Arrangement Dataset

1. We have a complex framed question in the start that can have multiple relations entangled with the entities in the same sentence.
2. Convert complex sentences into simpler sentences using the NER and RE framework, similar to the approach for family tree problems like shown in the step-2 of the Figure 9
3. Extract constraints (e.g., "A sits 2 places right of B") and convert them into a standard format using the RAG framework, as done with

family tree problems as shown in step-3 of Figure 9

4. Apply graph theory (DFS) to solve the problem using the extracted constraints to determine the circular arrangement using the constraint list to create an adjacency list and make a directed graph from it. For e.g. the constraint (A,B,right,2) will have an edge of +2 from node A to node B and an edge of -2 from node B to node A (since going from A to B is right and going from B to A is left).

This proposed pipeline should definitely work better for the seating arrangement dataset. We have seen that GPT-3.5-turbo struggles with complex, unstructured sentences but performs well with direct and structured sentences. By simplifying complex sentences using the NER and RE framework, and employing the RAG framework to retrieve relevant instructions, we can significantly enhance the model's ability to process and understand the data effectively.

## 7 Conclusion

This analysis study reveals critical insights on strengths and shortcomings in the domain of logical and positional reasoning tasks of a state-of-the-art large language model, GPT-3.5-turbo. The findings underline a significant disparity in the performance of the model across different types of reasoning problems.

On the one hand, GPT-3.5-turbo shows strong performance on the structured arithmetic reasoning task, with a test accuracy of 96.83% on the Multi-Arith dataset. This accuracy is significantly better than the 78.7% that earlier models, such as text-davinci002, have achieved on handling single-step reasoning tasks.

In contrast, the study shows very pronounced weaknesses of GPT-3.5-turbo in the face of more complex, real-world-like reasoning scenarios. Its performance drops down to 36.58% accuracy on the family relationship puzzle dataset and degrades further to 24.87% on the seating arrangement dataset. These kinds of tasks, which require elaborate logical relations and spatial reasoning, show clear difficulties that this model has in multi-hop reasoning, especially if the questions are not structured or require understanding concepts of relative directions that depend on the context.

The implications of these findings are significant

for the field of Natural Language Processing and Artificial Intelligence:

1. Despite the great strides in this direction, even state-of-the-art language models—like GPT-3.5-turbo—have significant limitations with tasks that require multi-step reasoning in complexity. This gap needs to be acknowledged and addressed in applications where such reasoning is critical.
2. The pipelines proposed with Named Entity Recognition, Relationship Extraction, and Retrieval-Augmented Generation are a pointer to how far one could go in augmenting language models using structured knowledge and reasoning techniques in a manner that emulates this combination. It follows an increasing trend in NLP regarding the integration of symbolic and neural approaches.
3. This study suggests several areas for future research, like improving multi-hop reasoning capabilities, developing better methods for handling spatial and relational reasoning, and creating superior evaluation datasets to capture real-world reasoning complexities.

## 8 Acknowledgements

I would like to express my sincere gratitude to Professor Ndapa Nakashole for her invaluable guidance and support throughout this project. Her insightful feedback and expertise have been instrumental in shaping the direction and quality of my work.

I would also like to extend my thanks to AI tools, particularly ChatGPT, for assisting in correcting grammatical errors in my report and rephrasing certain sentences to better articulate my objectives. The use of these tools has significantly enhanced the clarity and readability of my work.

## References

- Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M.-W. (2020). Realm: Retrieval-augmented language model pre-training.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2023). Large language models are zero-shot reasoners.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.



Roy, S. and Roth, D. (2015). Solving general arithmetic word problems. In Màrquez, L., Callison-Burch, C., and Su, J., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.