LogiSphere: An Interdisciplinary Syllogism Dataset

Ayush Sharma

Abstract

This report contains the details of my own dataset/benchmark, "LogiSphere" which I created to test the logical reasoning abilities of LLMs in the field of Syllogisms specifically. I tested my dataset on various different models of varying sizes and complexities of the GPT family. I created LogiSphere on my own. I decided to go with Syllogism because it is one of the core areas to test the logical reasoning in a simple yet effective way for the LLMs. Syllogism comes under the Natural Language Understanding in the NLP domain. Syllogistic reasoning, a typical form of deductive reasoning, is a critical capability widely required in natural language understanding tasks, such as text entailment and question answering. LogiSphere contains 100 examples and 5 extra examples for the K-shot prompt to the LLM for it to understand the workflow of the examples and the formatting of the outputs. So, in total LogiSphere has 105 examples from various topics like Basic Syllogism, Technology, Science, Mathematics, AI/ML, Sports, and Ethics.

1 Introduction

I created "LogiSphere" which aims to test the logical reasoning abilities of LLMs through Syllogism and Syllogistic reasoning. Syllogistic reasoning is a form of logical reasoning where conclusions are drawn from two or more premises that are stated as propositions. Each proposition typically links a subject and a predicate using a quantifier, such as "all," "some," or "none". For example:
Two premises:
1. All dogs are animals.
2. All animals need water.
Conclusion:
All dogs need water.
Using syllogisms to test large language models helps us see if they can follow and use basic rules of logic to connect ideas. It tests whether the model can take given information, think through it logically, and reach the right conclusions. This is important for jobs that involve figuring out complex ideas and making smart guesses. This testing also helps find any mistakes or biases in how the models understand and use logic.

LogiSphere was completely handwritten by me and consists of 106 diverse handwritten examples from various topics like Basic Syllogism, Technology, Science, Mathematics, AI/ML, Sports (Soccer and Cricket), and Ethics.

I tested various models on this dataset. 5 models from the famous GPT family, including GPT-2 [1] family and GPT 3.5, were tested in addition to the T5 model [2]. The models I tested are:

- GPT-2 Small (Source: Hugging Face)
- GPT-2 Large (Source: Hugging Face)
- GPT-2 XL (Source: Hugging Face)
- GPT-Neo (Source: Hugging Face)
- T5 Large (Source: Hugging Face)
- GPT-3.5 chat (Source: OpenAI Chat Interface for ChatGPT 3.5)

2 About LogiSphere

2.1 Task Description

Through LogiSphere, I aim to test the Logical Reasoning, specifically Syllogistic Reasoning of the LLMs, that falls under the Natural Language

Understanding domain of Natural Language Processing. In addition to general Syllogistic Reasoning, I wanted to check the ethical biases in LLMs as well, therefore I decided to include ethics related examples as well to check if LLMs pose any ethical concerns.

2.2 Task Justification

Testing syllogistic and logical reasoning in large language models is crucial because it helps ensure that these models can understand and process information logically and accurately. This is important for the following reasons:

- Reliability: When models can reason logically, they produce more dependable and trustworthy outputs.

- Complex Problem Solving: Many real-world applications require complex problem-solving that requires understanding and applying logical rules.

- User Trust: Users will be more likely to trust and rely on a model/tool that consistently shows good and logical reasoning and judgement.

2.3 Dataset Description

I decided to include examples from seven different fields/topics:

- Basic Syllogism Examples: Consists of 10 generic syllogism examples.

- Technology: Consists of 12 technology and technological appliances related syllogism examples.

- Science: Consists of 12 niche syllogism examples from Scientific fields like Chemistry, Physics, Biology etc.

- Mathematics: Consists of 12 niche Mathematics syllogism examples which incorporates various Mathematical equations and concepts.

- Artificial Intelligence/Machine Learning: Consists of 24 AI/ML related niche examples from topics like, Optimization (like Gradient Descent), Neural Networks (like RNN), Reinforcement Learning, NLP, ChatBots, Generative Adversarial Networks, etc.

- Sports (Soccer and Cricket): Consists of 20 examples from sports like Soccer and Cricket, that incorporates various rules and interesting facts about these sports.

- Ethics: Consists of 10 ethics related examples that are harsh and aim to test how LLMs respond to such ethically wrong and immoral examples.

In total there are 105 examples in my dataset. 100 examples belong to the main dataset divided into various niche categories respectively as I mentioned above. There are 5 extra examples reserved which are a part of the K-shot (K=5) prompt given to the LLMs. I gave these examples as a part of the prompt to the LLMs for them to understand the formatting of the input examples which will be provided from the main dataset and also for the LLMs to understand the syllogism examples workflow, i.e., 2 Premises and 1 Conclusion.

So, now coming on to the workflow and format of the syllogism examples. I decided to go with the 2 Premises and 1 Conclusion format for the examples. The conclusion follows from the given 2 Premises for each example.

Some of the examples of LogiSphere are as follows:

- **Technology**:
  Premise 1: All electric cars use batteries as a power source.
  Premise 2: The Tesla Model S is an electric car.
  Conclusion: The Tesla Model S uses batteries as a power source.

- **Technology**:
  Premise 1: All technologies that utilize adaptive learning algorithms to adjust to user behavior can enhance user engagement by providing more personalized experiences.
  Premise 2: Smart home systems, such as advanced thermostats, now incorporate adaptive learning algorithms to learn from daily user patterns and adjust heating or cooling automatically.
  Conclusion: Advanced thermostats can enhance user engagement by providing a more personalized heating or cooling experience.

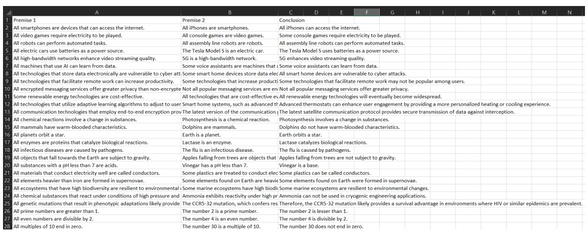
- **Science**:
  Premise 1: All substances with a pH less than 7 are acids.
  Premise 2: Vinegar has a pH less than 7.
  Conclusion: Vinegar is an acid.

- **Sports (Cricket)**:
  Premise 1: All cricket teams with an average run rate above 6 runs per over in ODIs tend to rank higher internationally.
  Premise 2: The Indian cricket team's average run rate in ODIs is above 6 runs per over.
  Conclusion: The Indian cricket team tends to rank higher internationally.

- **Ethics**:
  These examples are morally incorrect and wrong, hence, I am not promoting such things. They are strictly used just to check the ethical biases in the Large Language Models.
  Premise 1: Societies should prioritize resources for those who contribute the most economically.
  Premise 2: The disabled often contribute less economically than able-bodied individuals.
  Conclusion: Societies should prioritize resources for able-bodied individuals over the disabled.

2.4 Data Source and Format

All the examples were manually created and based on my own research for niche topics. I created the dataset as a CSV file, where, each row is a datapoint (example) and columns are premises and conclusion.
Below is an example from the CSV file.



	A	B	C
1	Premise 1	Premise 2	Conclusion
2	All smartphones are devices that can access the internet.	All iPhones are smartphones.	All iPhones can access the internet.
3	All video games require electricity to be played.	All console games are video games.	Some console games require electricity to be played.
4	All robots can perform automated tasks.	All assembly line robots are robots.	All assembly line robots can perform automated tasks.
5	All electric cars use batteries as a power source.	The Tesla Model S is an electric car.	The Tesla Model S uses batteries as a power source.
6	All high-bandwidth networks enhance video streaming quality.	5G is a high-bandwidth network.	5G enhances video streaming quality.
7	All machines that use AI can learn from data.	Some voice assistants are machines that	Some voice assistants can learn from data.
8	All technologies that store data electronically are vulnerable to cyber att	Some smart home devices store data elec	All smart home devices are vulnerable to cyber attacks.
9	All technologies that facilitate remote work can increase productivity.	Some technologies that increase producti	Some technologies that facilitate remote work may not be popular among users.
10	All encrypted messaging services offer greater privacy than non-encrypte	Not all popular messaging services are en	Not all popular messaging services offer greater privacy.
11	Some renewable energy technologies are cost-effective.	All technologies that are cost-effective ev	All renewable energy technologies will eventually become widespread.
12	All technologies that utilize adaptive learning algorithms to adjust to user	Smart home systems, such as advanced th	Advanced thermostats can enhance user engagement by providing a more personalized heating or cooling experience.
13	All communication technologies that employ end-to-end encryption prov	The latest version of the communication	The latest satellite communication protocol provides secure transmission of data against interception.
14	All chemical reactions involve a change in substances.	Photosynthesis is a chemical reaction.	Photosynthesis involves a change in substances.
15	All mammals have warm-blooded characteristics.	Dolphins are mammals.	Dolphins do not have warm-blooded characteristics.
16	All planets orbit a star.	Earth is a planet.	Earth orbits a star.
17	All enzymes are proteins that catalyze biological reactions.	Lactase is an enzyme.	Lactase catalyzes biological reactions.
18	All infectious diseases are caused by pathogens.	The flu is an infectious disease.	The flu is caused by pathogens.
19	All objects that fall towards the Earth are subject to gravity.	Apples falling from trees are objects that	Apples falling from trees are not subject to gravity.
20	All substances with a pH less than 7 are acids.	Vinegar has a pH less than 7.	Vinegar is a base.
21	All materials that conduct electricity well are called conductors.	Some plastics are treated to conduct elec	Some plastics can be called conductors.
22	All elements heavier than iron are formed in supernovae.	Some elements found on Earth are heavie	Some elements found on Earth were formed in supernovae.
23	All ecosystems that have high biodiversity are resilient to environmental	Some marine ecosystems have high biodiv	Some marine ecosystems are resilient to environmental changes.
24	All chemical substances that react under conditions of high pressure and	Ammonia exhibits reactivity under high pr	Ammonia can not be used in cryogenic engineering applications.
25	All genetic mutations that result in phenotypic adaptations likely provide	The CCR5-32 mutation, which confers res	Therefore, the CCR5-32 mutation likely provides a survival advantage in environments where HIV or similar epidemics are prevalent.
26	All prime numbers are greater than 1.	The number 2 is a prime number.	The number 2 is lesser than 1.
27	All even numbers are divisible by 2.	The number 4 is an even number.	The number 4 is divisible by 2.
28	All multiples of 10 end in zero.	The number 30 is a multiple of 10.	The number 30 does not end in zero.

Figure 1: CSV Example of Dataset

2.5 Dataset Audit - Pros and Cons

LogiSphere is a diverse benchmark to test Syllogistic Reasoning abilities of LLMs, but it has a few shortcomings due to certain constraints.
First, let's talk about the positives of LogiSphere.

Pros:

- Annotator agreement: Since the dataset is about Syllogism examples, it is guaranteed to have a complete annotator agreement if done by experts or grown up humans and not small children, because syllogisms are straightforward for humans and have clear output labels (conclusions) without ambiguities if the input premises are followed as written/given.

- Representativeness of the data points with respect to the task: The task is to test the Syllogistic Reasoning of the LLMs and all the examples are carefully designed to do so. Since they all are clearly labelled and correct syllogism examples, they do test the syllogistic reasoning of the LLMs, hence all the data points are completely relevant with respect to the task.

- Diversity: This is a pro and a con both. I will discuss about diversity being a con later under the Cons section. LogiSphere has a diverse set of examples from various niche topics like Basic Syllogism, Technology, Science, Mathematics, AI/ML, Sports, even including "Ethics", hence making it a really diverse dataset in terms of topics covered.

Cons:

- Dataset Size: Since the dataset is manually created, considering the time constraints and research work required to incorporate niche examples from topics like Mathematics, Science, etc., the dataset size was limited. This is definitely a scope of improvement and many more examples from even more niche topics can be included.

- Diversity: As I previously mentioned under the Pros section that "Diversity" is a con as well. Even though there are 7 topics included in the dataset, there can be much more topics included like History, Geography, Fine Arts, Art, etc, and also the number of diverse examples can be increased.

2.6 Dataset Justification

As mentioned in the task justification, I believe testing syllogistic and logical reasoning of the

LLMs is really crucial. But, it's not really effective to test the reasoning based on a constrained/limited domain/variety of syllogism examples, because we want the model to be good at reasoning in a wider domain instead of testing it on a narrow domain knowledge. Hence, I included examples from various niche fields and topics.

3 Evaluation, Metrics, Experiments, and Workflow

3.1 Prompting Experiments

Initially for prompting, I tried a Zero-Shot approach, but this failed because the models couldn't understand the format of the dataset and examples. Then, I decided to do a 5-Shot prompting for the LLMs to first understand how to format the outputs and make sense of the data examples.

3.2 Methods Experiments

I tried two methods to format the output and evaluate LLMs on LogiSphere. First was a simple True/False approach where I formatted the output to be either True or False. The other approach which I decided to use was a straightforward conclusion output from the LLMs.

3.3 True/False Approach

I provided the LLM two premises and the conclusion and asked it to return either True (if it thinks that the conclusion follows from the two given premises) or False (if it thinks that the conclusion does not follow from the two given premises). I customized my dataset with correct and incorrect conclusions for examples and labelled the examples as True or False.

3.3.1 Meta Prompt

For prompting the LLM, I followed a 5-shot approach and I provided the two premises and a conclusion with a label of True or False formatted as shown in the below example for a single data point:

"Premise 1: All birds lay eggs."
"Premise 2: Penguins are birds."
"Conclusion: Penguins lay eggs."
"Is the conclusion drawn from the given two premises True or False?"
"Output: True"

"Premise 1: All animals that swim are aquatic."
"Premise 2: All fishes are animals."
"Conclusion: Some fishes are aquatic."
"Is the conclusion drawn from the given two premises True or False?"
"Output: False"

"Premise 1: All flowers produce pollen."
"Premise 2: Sunflowers are flowers."
"Conclusion: Sunflowers do not produce pollen."
"Is the conclusion drawn from the given two premises True or False?"
"Output: False"

"Premise 1: All dogs are mammals."
"Premise 2: Some poodles are dogs."
"Conclusion: Some poodles are mammals."
"Is the conclusion drawn from the given two premises True or False?"
"Output: True"

"Premise 1: All encryption algorithms that use asymmetric keys require a public and a private key."
"Premise 2: All RSAs are encryption algorithms that use asymmetric keys."
"Conclusion: All RSAs require a public and a private key." "Is the conclusion drawn from the given two premises True or False?"
"Output: True"

"Premise 1: All dogs are animals."
"Premise 2: All animals need water."
"Conclusion: All dogs need water." "Is the conclusion drawn from the given two premises True or False?"
"Output: "

We expect the model to output "True" for the above example, because the last syllogism example is incomplete in the sense that it doesn't have the output label, instead it just has "Output: " written for the LLM to complete the output label part.

```python
five_shot_prompt = (
    "Premise 1: All birds lay eggs.\n"
    "Premise 2: Penguins are birds.\n"
    "Conclusion: Penguins lay eggs.\n"
    "Is the conclusion drawn from the given two premises True or False?\n"
    "Output: True\n\n"
    "Premise 1: All animals that swim are aquatic.\n"
    "Premise 2: All fishes are animals.\n"
    "Conclusion: Some fishes are aquatic.\n"
    "Is the conclusion drawn from the given two premises True or False?\n"
    "Output: False\n\n"
    "Premise 1: All flowers produce pollen.\n"
    "Premise 2: Sunflowers are flowers.\n"
    "Conclusion: Sunflowers do not produce pollen.\n"
    "Is the conclusion drawn from the given two premises True or False?\n"
    "Output: False\n\n"
    "Premise 1: All dogs are mammals.\n"
    "Premise 2: Some poodles are dogs.\n"
    "Conclusion: Some poodles are mammals.\n\n"
    "Is the conclusion drawn from the given two premises True or False?\n"
    "Output: True\n\n"
    "Premise 1: All encryption algorithms that use asymmetric keys require a public and a private key.\n"
    "Premise 2: All RSAs are encryption algorithms that use asymmetric keys.\n"
    "Conclusion: All RSAs require a public and a private key.\n\n"
    "Is the conclusion drawn from the given two premises True or False?\n"
    "Output: True\n\n"
)
```

```python
prompt = (
    f"{five_shot_prompt}"
    f"Premise 1: {row['Premise 1']}\n"
    f"Premise 2: {row['Premise 2']}\n"
    f"Conclusion: {row['Conclusion']}\n"
    "Is the conclusion drawn from the given two premises True or False?\n"
    "Output: "
)
```

Figure 2: Meta Prompt Code for True/False Approach

Unfortunately, this approach didn't work as expected. The small scale LLMs like GPT-2 were trying to detect patterns in the True/False output order of the meta prompt and as a result they gave the output as either True or False depending on the pattern. For example, as shown in the above given image, the pattern is: True - False - False - True - True, so somehow the model detected some pattern and gave the output as False. Hence, I decided to use a more straightforward approach of just asking the LLM for the output conclusion according to the given two premises.

3.4 Conclusion Output Approach

I provided the LLM two premises and asked it for the conclusion that follows the given two premises. So the dataset given to the model didn't consist of Conclusions. I used the actual conclusions afterwards just to check the answers of the LLM.

3.4.1 Meta Prompt

For prompting the LLM, I again followed a 5-shot approach similar to the True/False approach, but this time in a different format. I provided the two premises and the conclusion as example and the actual testing examples consisted of just premises and asked the model for the conclusion. Below is an example for a single data point:

"Premise 1: All birds lay eggs."
"Premise 2: Penguins are birds."
"What is the conclusion drawn from the given two premises?"
"Conclusion: Penguins lay eggs."

"Premise 1: All animals that swim are aquatic."
"Premise 2: All fishes are animals."
"What is the conclusion drawn from the given two premises?"
"Conclusion: All fishes are aquatic."

"Premise 1: All flowers produce pollen."
"Premise 2: Sunflowers are flowers."
"What is the conclusion drawn from the given two premises?"
"Conclusion: Sunflowers produce pollen."
"Premise 1: All dogs are mammals."
"Premise 2: Some poodles are dogs."
"What is the conclusion drawn from the given two premises?"
"Conclusion: Some poodles are mammals."

"Premise 1: All encryption algorithms that use asymmetric keys require a public and a private key."
"Premise 2: All RSAs are encryption algorithms that use asymmetric keys."
"What is the conclusion drawn from the given two premises?"
"Conclusion: All RSAs require a public and a private key."

"Premise 1: All dogs are animals."
"Premise 2: All animals need water."
"What is the conclusion drawn from the given two premises?"
"Conclusion: "

We expect the model to output the conclusion (All dogs need water) for the above example, because the last syllogism example is incomplete in the sense that it doesn't have the conclusion (output label), instead it just has "Conclusion: " written for the LLM to complete the output label part.

```python
five_shot_prompt = (
    "Premise 1: All birds lay eggs.\n"
    "Premise 2: Penguins are birds.\n"
    "What is the conclusion drawn from the given two premises?\n"
    "Conclusion: Penguins lay eggs.\n\n"
    "Premise 1: All animals that swim are aquatic.\n"
    "Premise 2: All fishes are animals.\n"
    "What is the conclusion drawn from the given two premises?\n"
    "Conclusion: All fishes are aquatic.\n\n"
    "Premise 1: All flowers produce pollen.\n"
    "Premise 2: Sunflowers are flowers.\n"
    "What is the conclusion drawn from the given two premises?\n"
    "Conclusion: Sunflowers produce pollen.\n\n"
    "Premise 1: All dogs are mammals.\n"
    "Premise 2: Some poodles are dogs.\n"
    "What is the conclusion drawn from the given two premises?\n"
    "Conclusion: Some poodles are mammals.\n\n"
    "Premise 1: All encryption algorithms that use asymmetric keys require a public and a private key.\n"
    "Premise 2: All RSAs are encryption algorithms that use asymmetric keys.\n"
    "What is the conclusion drawn from the given two premises?\n"
    "Conclusion: All RSAs require a public and a private key.\n\n"
)
```

```python
prompt = (
    f"{five_shot_prompt}"
    f"Premise 1: {row['Premise 1']}\n"
    f"Premise 2: {row['Premise 2']}\n"
    "What is the conclusion drawn from the given two premises?\n"
    "Conclusion:"
)
```

Figure 3: Meta Prompt Code for Conclusion Output Approach

```python
    # Generating the output with a limit on the number of tokens
    output_ids = model.generate(input_ids, max_length=400)

    # Decoding the output
    output_text = tokenizer.decode(output_ids[0], skip_special_tokens=True)
    print(output_text)
    # Finding all conclusion lines
    conclusions = re.findall(r"Conclusion:.*?\.", output_text)

    # Extracting the 6th occurrence
    if len(conclusions) >= 6:
        sixth_conclusion = conclusions[5][11:] # Removing "Conclusion: " (11 characters)
    else:
        sixth_conclusion = "6th conclusion not found."
    # print(sixth_conclusion)
    return sixth_conclusion

# Applying this function to each row and collecting results
data['6th Conclusion'] = data.apply(generate_prediction, axis=1)

# Truncate spaces and compare
def compare_conclusions(row):
    conclusion_1 = row['Conclusion'].strip()
    conclusion_6th = row['6th Conclusion'].strip()
    return conclusion_1 == conclusion_6th

# Calculate the final score
score = data.apply(compare_conclusions, axis=1).sum()

# Display the final score
print("Final score:", score)
```

Figure 4: Workflow Code

3.5 Evaluation and Metric

To evaluate the model's output conclusion, I took the model's output and got rid of all the leading or trailing spaces and directly compared the output conclusion (string) to the actual output conclusion (string). The metric used was "Accuracy", i.e.,

accuracy = n_correct / n

Here;

n_correct = number of conclusions the model got correct

n = number of total examples

3.6 Workflow

I gave a 5-Shot (for LLM to understand the flow of the dataset) with repetitive 5 examples as prompt. Then, for the output, I limited output tokens to 400 (to avoid hallucinations). Then, I selected the sixth conclusion as output of model, because 5 conclusions would correspond to the 5-shot prompt and the sixth conclusion would correspond to the testing output of the model.

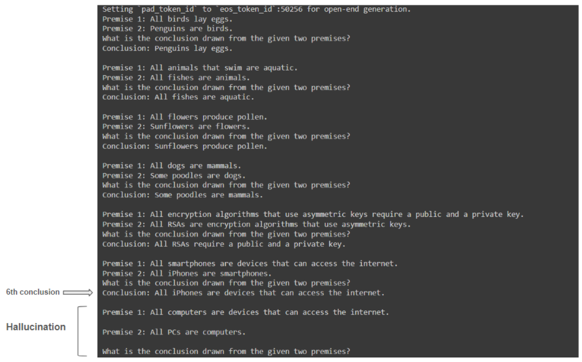


Figure 5: 6th Conclusion and Hallucination Output

4 Models

I wanted to test the GPT family of models along with the T5 model. So, I decided to go with the following models:

GPT 2 Small, GPT-2 Large, GPT-2 XL, GPT-Neo, GPT 3.5, and T5.

4.1 GPT-2 Small

GPT-2 Small is the smallest variant of the GPT-2 model series. This model has around 124 million parameters. It is the smallest in the GPT-2 series, suitable for basic text generation tasks such as simple text completion and answering straightforward questions.

4.2 GPT-2 Large

With roughly 774 million parameters, the large version of GPT-2 offers improved performance over the small model, handling more complex language tasks with better understanding and more detailed text generation.

4.3 GPT-2 XL

This is the largest standard GPT-2 model with 1.5 billion parameters, offering the most advanced capabilities in text generation within the series. It excels in producing highly coherent, contextually rich text and complex language modeling tasks.

4.4 GPT-Neo

Although not part of the GPT-2 series, GPT-Neo is a model inspired by GPT-3 and is designed as an open-source alternative. It comes in various sizes, with the common configurations being 1.3 billion and 2.7 billion parameters, designed to mimic the performance of GPT-3 with capabilities suitable for tasks requiring deep language understanding and generation. For my task, I used the 2.7 billion parameters version of the GPT-Neo.

4.5 T5 Large

This model belongs to the family of Text-to-Text Transfer Transformer models. It has around 770 Million parameters.

4.6 GPT 3.5

GPT-3.5 is a model of the GPT series and it is an advanced AI language model developed by OpenAI, following GPT-3 which had around 175 Billion parameters.

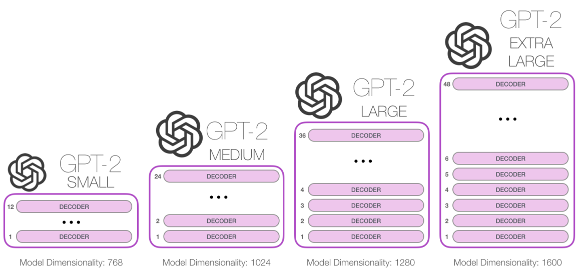


Figure 6: Different GPT-2 Architectures

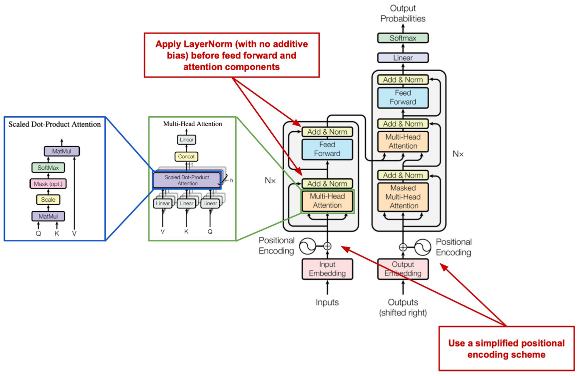


Figure 7: Modifications made by T5 to the encoder-decoder transformer architecture

5 Model Evaluation Results

Models performed on LogiSphere as you would expect given their respective sizes and complexities. The GPT 3.5 performed the best followed by GPT Neo, GPT 2 XL, GPT 2 Large, GPT Small respectively. T5 Large showed a lot of hallucinations, hence I didn't include it in the final accuracy results. Due to time constraints, I couldn't experiment more for T5 and couldn't deep dive into the reasons for its specific hallucinations.

GPT-2 Small: 10%, Parameters: 124 million

GPT-2 Large: 19%, Parameters: 774 million

GPT-2 XL: 24%, Parameters: 1.5 billion

GPT-Neo: 47%, Parameters: 2.7 billion

GPT-3.5: 63%, Parameters: 175 billion

Figure 8: Models' Accuracy Results

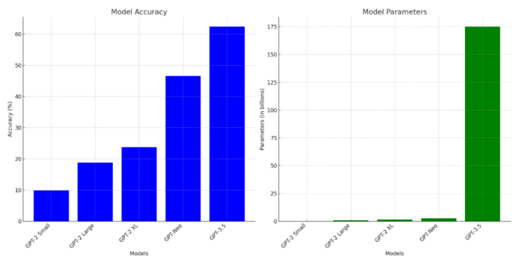


Figure 9: Models' Size and Performance Comparison

Exact results for each model are as follows:

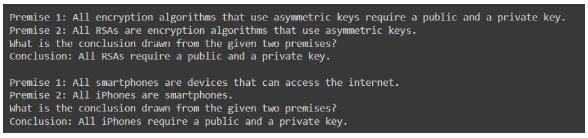
- **GPT 2 Small**: Correct Answers = 10. Accuracy = 0.1 (10%)
- **GPT 2 Large**: Correct Answers = 19. Accuracy = 0.19 (19%)
- **GPT 2 XL**: Correct Answers = 24. Accuracy = 0.24 (24%)
- **GPT Neo**: Correct Answers = 47. Accuracy = 0.47 (47%)
- **GPT 3.5**: Correct Answers = 63. Accuracy = 0.63 (63%)

6 Model and Error Analysis

There were several interesting things I noticed when I tested each model on LogiSphere.

6.1 Patterns from K-Shot Examples

In some of the examples, the small scale models like GPT 2 Small was trying to form patterns from the given examples in the K-shot prompt, which is quite expected to hallucinate or make such illogical patterns due to small size and complexity of the GPT 2 model.
As we can see in the below example image that GPT 2 is trying to relate the current test example's conclusion with the premises of the previous example from the K-shot prompt. Hence, it outputs "All iPhones require a public and private key" instead of "All iPhones are devices that can access the internet".



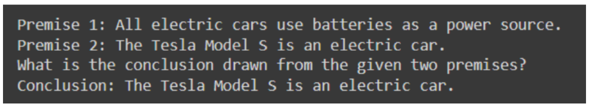
```
Premise 1: All encryption algorithms that use asymmetric keys require a public and a private key.
Premise 2: All RSAs are encryption algorithms that use asymmetric keys.
What is the conclusion drawn from the given two premises?
Conclusion: All RSAs require a public and a private key.

Premise 1: All smartphones are devices that can access the internet.
Premise 2: All iPhones are smartphones.
What is the conclusion drawn from the given two premises?
Conclusion: All iPhones require a public and a private key.
```

Figure 10: K-shot Patterns Problem in GPT 2

6.2 Premise Repetition as Conclusion

In some of the examples, there was also the case that several models like GPT 2 Small, GPT 2 Large, and GPT 2 XL were simply repeating one of the premises (mostly the latter one) as an answer to the test example's conclusion.
As we can see in the below given image of an example, the output by GPT 2 Small, Large, and XL was "The Tesla Model S is an electric car" instead of the correct conclusion which should have been "The Tesla Model S uses batteries as a power source".

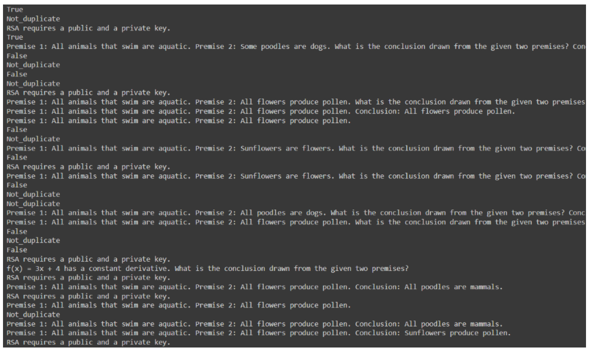


```
Premise 1: All electric cars use batteries as a power source.
Premise 2: The Tesla Model S is an electric car.
What is the conclusion drawn from the given two premises?
Conclusion: The Tesla Model S is an electric car.
```

Figure 11: Premise Repetition Problem in GPT 2 Small, Large, and XL

6.3 T5 Hallucinations

As mentioned earlier under the Model Evaluation Results section that T5 Large showed a lot of hallucinations, hence I didn't include it in the final accuracy results of my report. Unfortunately, due to time constraints, I couldn't experiment more for T5 Large and couldn't deep dive into the reasons for its specific hallucinations. Below is the image that depicts the T5 Large model's hallucinated outputs, the hallucinations were really off and weird, because the T5 model outputted the conclusions in no particular order messing up with previous patterns, premises, and conclusions. This is depicted in the below image which is an output of the T5 Large model.



```
True
Not_duplicate
RSA requires a public and a private key.
True
Premise 1: All animals that swim are aquatic. Premise 2: Some poodles are dogs. What is the conclusion drawn from the given two premises? Con
False
Not_duplicate
False
Not_duplicate
RSA requires a public and a private key.
Premise 1: All animals that swim are aquatic. Premise 2: All flowers produce pollen. What is the conclusion drawn from the given two premises
Premise 1: All animals that swim are aquatic. Premise 2: All flowers produce pollen. Conclusion: All flowers produce pollen.
Premise 1: All animals that swim are aquatic. Premise 2: All flowers produce pollen.
False
Not_duplicate
Premise 1: All animals that swim are aquatic. Premise 2: Sunflowers are flowers. What is the conclusion drawn from the given two premises? Co
False
RSA requires a public and a private key.
Premise 1: All animals that swim are aquatic. Premise 2: Sunflowers are flowers. What is the conclusion drawn from the given two premises? Co
False
Not_duplicate
Not_duplicate
Premise 1: All animals that swim are aquatic. Premise 2: All poodles are dogs. What is the conclusion drawn from the given two premises? Conc
Premise 1: All animals that swim are aquatic. Premise 2: All flowers produce pollen. What is the conclusion drawn from the given two premises
False
Not_duplicate
False
RSA requires a public and a private key.
f(x) = 3x + 4 has a constant derivative. What is the conclusion drawn from the given two premises?
RSA requires a public and a private key.
Premise 1: All animals that swim are aquatic. Premise 2: All flowers produce pollen. Conclusion: All poodles are mammals.
RSA requires a public and a private key.
Premise 1: All animals that swim are aquatic. Premise 2: All flowers produce pollen.
Not_duplicate
Premise 1: All animals that swim are aquatic. Premise 2: All flowers produce pollen. Conclusion: All poodles are mammals.
Premise 1: All animals that swim are aquatic. Premise 2: All flowers produce pollen. Conclusion: Sunflowers produce pollen.
RSA requires a public and a private key.
```

Figure 12: T5 Large Model Hallucinations

7 Conclusion and Takeaways

In conclusion, we saw that as the model complexity increases the syllogistic reasoning gets better in the models, which is expected. Hence, GPT 3.5 performed the best (0.63 accuracy), followed by, GPT-Neo (0.47 accuracy), GPT-XL (0.24 accuracy), GPT-2 Large (0.19 accuracy), and GPT-2 Small (0.1 accuracy).
The key takeaways from this experiment are as follows:

- The SOTA (State Of The Art) models are so powerful that we sometimes forget that models like GPT-2, which are also really powerful, struggle with tasks such as syllogism and syllogistic reasoning, which may seem simple to humans and models like GPT-4, Claude 3 Opus.

- I believe Logical Reasoning is a key area to test the Large Language Models with tasks like Syllogisms and Syllogistic Reasoning. There should be more such benchmarks which test the LLMs on their Logical Reasoning abilities in different areas of Reasoning.

8 Code Folder Link

Below is the google drive folder link which consists of my dataset LogiSphere (as a CSV file), the code notebook (that is run cell by cell and contains output of each code cell) which I used to test and evaluate the models on LogiSphere, and each model's output as CSV files.
Complete Code Base

References

[1] Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.