ScenarioQA: Evaluating Test Scenario Reasoning Capabilities of Large Language Models

Shreya Sinha
Electrical and Computer Engineering Department
University of California, Santa Cruz
Santa Cruz, CA
ssinha12@ucsc.edu

Ishaan Paranjape
Computational Media Department
University of California, Santa Cruz
Santa Cruz, CA
iparanja@ucsc.edu

Jim Whitehead
Computational Media Department
University of California, Santa Cruz
Santa Cruz, CA
ejw@ucsc.edu

***Abstract*—Autonomous Vehicles (AVs) have the potential of reducing car accidents and increasing accessibility to transportation. AVs need to be rigorously tested. Scenario-based testing offers a set of approaches to design high-risk tests for AVs at low cost. Since the AVs need to be tested for a large number of scenarios, automated generation approaches are needed. Pre-trained Large Language Models (LLMs) are open-input, general-purpose data generators with good learning and reasoning abilities. However, due to the black-box nature of these systems, it's difficult to get direct evidence of their abilities. In this paper, we address the open question of the reasoning capabilities of pre-trained LLMs specifically in the context of scenario-based testing of AVs. Inspired by QA benchmarks for LLM evaluations for commonsense reasoning, science reasoning, and more, we present our main contribution, *ScenarioQA*. This benchmark involves an LLM-based QA generation process based on an integration of methods to generate questions and corresponding answers specifically in the context of scenario-based testing. We carry out a comprehensive evaluation of this process and gain valuable insights regarding effective QA generation. In addition, we evaluate several available pre-trained LLMs for these abilities.**

***Index Terms*—scenario-based testing, autonomous vehicles, large language models, artificial intelligence**

I. Introduction

Autonomous Vehicles (AVs) have the potential to eliminate accidents created by human error and reduce traffic fatalities by up to 90% [1]. In addition, AVs may even allow for accessible transportation for people with disabilities [2] as well as reducing environmental impact through innovative designs, enhancement of traffic flow, and technical advancements [3]. Due to the large number of sensors and a complex technology stack incorporated in AVs, it's necessary to test their behavior before releasing them for general public use to ensure that they react safely under given circumstances. The most prevalent approach used for testing is creating simulations of the driving environment [4]. Many different paradigms for testing are considered including real-world, shadow modem simulation, hybrid, XIL, etc.

Simulation in particular allows testing under different conditions and environments (both with static and dynamic elements) in the form of scenarios. Particularly, scenarios allow for the creation of high-risk, low-cost environments [5] which is difficult to design and carry out as a part of real-world testing. Real-world field tests provide further insight under real driving conditions and are complementary to simulation testing [5]. Mostly, scenarios are hand-authored (such as in CARLA Scenario Runner or Scenic), as such, automated generation of scenarios is necessary since AVs need to be tested for a large number of test scenarios. Large Language Models (LLMs) are *open-input*, general purpose data generators that have been gaining significance recently with the advent of tools such as *ChatGPT*. A comprehensive understanding of the capabilities of LLMs to help with automated scenario generation remains an open question. In this paper, we present an initial evaluation to gain this understanding across various pre-trained LLMs and prompting methods.

LLMs are neural network models for textual data. They can generate a wide range of data. [6]. A few advantages are the capability of in-context learning and reasoning as a result of which they are capable of arriving at solutions to problems without pre-written instructions [7]. One disadvantage is that they are black box models where the general reasoning process to conclude something is inaccessible, therefore any decisions or conclusions that the network reaches could be scrutinized due to the unpredictability of the LLM's behavior. Therefore, we aim to focus our evaluation on reasoning evaluations within the context of scenario-based testing of AVs.

One approach for creating more transparency in evaluating the capabilities of LLMs is question-answering benchmarks. Textual Question Answering (QA) [8][9][10] papers use unstructured data to provide precise answers to users' questions in natural language processing [11]. QA benchmarks can be used to provide evaluations for users to understand the reasoning of LLMs. **Our contribution is that we adapt and extend several current QA approaches to create a new QA benchmark for evaluating the reasoning capabilities of pre-trained LLMs within the context of scenario-based testing of AVs.** In addition, to scale the number of questions available and to coherently compose questions with multiple structures, we make use of a pre-trained LLM. One implication of this then can be the design of pre-trained LLM-based scenario generation tools.

The rest of the paper is divided into sections as follows:

In section 2, we go over related works, look at other QA sources, and determine questions that will evaluate the specific reasoning capabilities of GPT. In section 3, we present an approach for formulating questions and an automated process for scaling question generation using GPT-4. In section 4, we present the design of our experiments. Section 5 contains a summary of the results and a discussion of the same. Section 6 concludes this paper and presents some ideas regarding future work. The code for this paper is located at the following URL: https://github.com/AugmentedDesignLab/ScenarioQA

II. Related Works

A. Scenario Based Testing of AVs:

Before vehicles with high levels of automation such as Level 3 (conditionally automated) or Level 4 (high automation) [12] can be put on the market for commercial use, they must be thoroughly tested with an efficient assessment [13]. Many such tests exist including real-world testing, function-based testing, and even shadow mode, but one of the most promising and effective methods is the scenario-based testing approach. According to [13] a scenario can be classified as a temporal sequence of actions or events that occur in a scene such as a road or intersection with multiple participants and objects all playing a role in the simulation. Scenarios can help to support the development process by developing the necessary software and hardware components as well as testing the safety features of those components and ensuring their reliability [12].

Often creating and testing a scenario is a meticulous and time-intensive process and there is a necessity to specify all scenario details and define test scenarios manually [14]. As such, it is imperative to have the potential to automatically generate test scenarios that are based on real-world traffic rules and scenarios [14].

B. Prompting and QA benchmarks:

LLMs can play a potential role in the simulation for AVs due to the complex abilities of modern-day language models. For instance, on average, ChatGPT was 63.41% accurate under 10 different reasoning categories [15] with zero-shot learning. Chain of Thought (CoT) reasoning can help LLMs not only in the few-shot learning setting but also in the fine-tuning setting which improves model performance and reliability [16]. CoT reasoning along with the answer was found to improve the reasoning ability of models. Especially for GPT-3, this increases the result to 75.17% [16]. One interesting component that can very effectively test the knowledge base of an LLM is domain-based QA pairs. Effectively, the formulation of multiple-choice questions involving question concepts, answer concepts, and distractors is strategically designed to challenge models and require deeper understanding rather than relying on surface-level clues [8].

C. LLMs for Autonomous Driving:

Although AVs have thorough and well-tested modules that can handle a variety of situations, they may easily fail when they come across unpredictable cases or accidents. Given that LLMs have access to a wealth of knowledge across a variety of domains, autonomous driving systems could benefit from improved text prediction [17]. Additionally, due to their various reasoning capabilities, LLMs have the potential to generate and analyze low-level vehicle controls [17]. Moreover, LLMs have the potential to integrate human-like intelligence into AV systems [18] which include skills and reasoning such as spatial reasoning, pattern recognition, predictive reasoning, and object recognition and classification. Although there have been attempts to integrate human reasoning and knowledge into these AV systems, they lack deeper reasoning ability inherent to humans [18] which LLMs can help bridge.

III. Method

This section details the question-answer (QA) generation process. Pre-trained LLM GPT-4, available via the OpenAI's API is prompted to generate QA pairs by providing it information regarding different, relevant forms of reasoning, ontologies for scenario-based testing of AVs and formulating MCQs in general. An overview followed by design details for each *subprompt* mentioned above is written in the remainder of this section. Running GPT-4 with this prompt results in sets of 25 questions of a specific reasoning type. The sets have varying levels of difficulty among the questions while covering unique concepts within scenario-based testing of AVs.

A. Question-Answer Generation Overview

We create a framework for structuring questions to evaluate a pre-trained LLM for various types of reasoning. A complete overview of this process is shown in the figure below. Broadly, the framework includes the following elements: (1) Reasoning-specific question structure: We consider multiple reasoning evaluations (as mentioned in [15]) in the QA format that is relevant to scenario-based testing of autonomous vehicles. (2) Concepts: We consider the ASAM OpenXOntology (described in subsection D) and [19] for creating the final ontology. (3) General question formulation structure: We broadly consider elements of the structure for generating MCQs shown from [20]. This gives us some foundational, structural elements applicable to all MCQs.

B. Given Scenario

A detailed traffic scenario with the static elements involving road with clear weather, lighting, and road conditions as well as information about the vehicles and/or pedestrians. Using this scenario, we were able to create questions for specific reasoning types.

C. Reasoning Types

The following subsection presents this framework for each reasoning type along with some questions that result from it. These questions are then included in the representative dataset (only a representative set of questions). In the next section, we evaluate a set of pre-trained LLMs with specific prompting styles (such as chain-of-thought) for our question dataset.

In this paper, the question generation framework integrates a format for each reasoning category, multiple choice

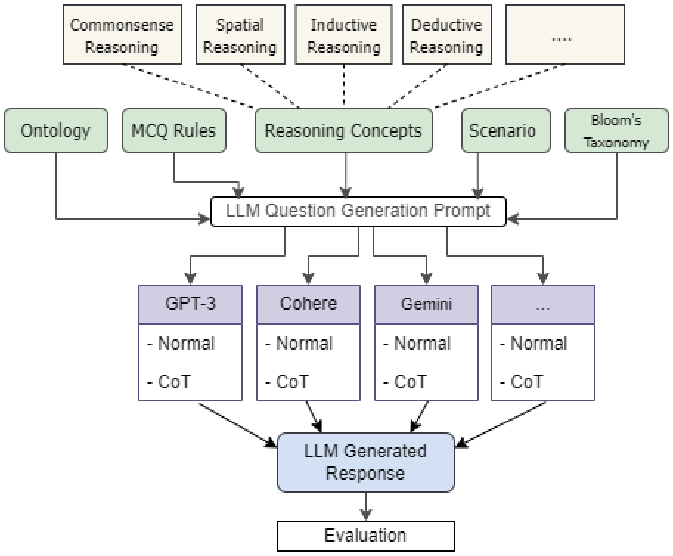


Fig. 1: Overall diagram for approach from conception to evaluation

questioning style as well as Bloom's taxonomy [20]. In addition, we provide an ontology that provides the necessary background concepts to the pre-trained LLM regarding scenario-based testing and examples of the reasoning type for reference which then provides adequate structure and details to the LLM to generate questions for each category. We will now detail question formats for each reasoning category below. This is followed by details regarding a general MCQ format which includes Bloom's taxonomy.

1) Commonsense Reasoning: Commonsense reasoning is used for understanding familiar knowledge and basic concepts to make predictions based on past behavior. CommonsenseQA is a benchmark for pre-trained LLMs to evaluate commonsense reasoning. It contains certain concepts and relations (sampled from ConceptNet) such as compositional or cause-effect relations [8]. These relations also indicate that commonsense skills are used in addressing the question. We maintain the relation and skill categories in this paper but instead use concepts from the scenario-based testing ontology detailed in this section further.

2) Deductive Reasoning: involves determining certain conclusions based on more general statements or assumptions [15]. For our purposes, EntailmentBank [15][9] has examples of subcategories of inferences that are used in deductive reasoning. We have referenced those subcategories in our framework to show the type of relation between each entity and to generate questions.

3) Inductive Reasoning: entails making predictions about new situations using previously known facts or existing knowledge. We have referenced the story generation method used by CLUTRR [10] as well as the snapshots of puzzles used to generate QA pairs based on inductive logic. These puzzles reference relations between people that are fairly easy for a human to determine, however, they may be a bit more challenging for an LLM to handle.

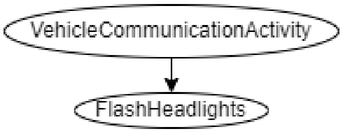


Fig. 2: An example ontology from the ASAM OpenXOntology standard

4) Spatial Reasoning: Spatial reasoning regards the ability to understand physical space given spatial relations among a few objects. A few language model evaluation benchmarks that involve spatial reasoning are detailed in [21] and [22]. The latter addresses the shortcomings of the former of being overly simplistic in terms of relations, given scene information and reasoning steps required. In this paper, given a scenario and the appropriate ontology, we evaluate spatial reasoning skills using spatial relations presented in SpartQA [22]. These are relative directions (such as left, right, above, below), qualitative distances (such as near and far) as well as crash-related spatial relations (such as near misses). In [22], reasoning takes place using *spatial rules* which resemble formal method relational properties such as transitivity and symmetry.

5) Temporal Reasoning: uses concepts such as frequency, duration, as well as relative ordering. In [23], complex temporal reasoning is evaluated by providing a multi-turn dialog followed by a multiple-choice sentence completion task that requires significant commonsense temporal reasoning to answer. In this paper, we consider the reasoning structure suggested by the reasoning categorization mentioned in [23]: general commonsense sense (such as the perception of *walking distance*), comparison (notions of *earlier* and *later* with respect to a given time) and arithmetic regarding time periods.

Using these aforementioned reasoning structures, we are able to evaluate the specific skills of pre-trained LLMs.

D. Scenario Ontology

We make use of the ASAM OpenXOntology [24]. Combining the conceptual structure of the ontology with structures for question formation and reasoning evaluations mentioned above, we can formulate effective questions. This ontology consists of two key sub-ontologies: core, domain, and application level. The core ontology refers to general concepts that are not necessarily about scenarios e.g. types of relations, states, and objects. Other ontologies are based on the core ontology. For this paper, we are focused on the domain ontology. This ontology provides us with standardized concepts and relations regarding scenario elements in each layer of the six-layer model [25]: roads, permanent road objects, temporary construction artifacts, vehicles, pedestrians, and environmental conditions. An example of a set of classes in this ontology is shown in figure 2. In addition, to *refine* this ontology, we consider the approach in [19].

E. General question formation structure

We consider the multiple choice question formation structure provided in [20]. Here, MCQs are generated using GPT which follows the learning objectives of a programming course. Here, an MCQ is simply defined as a question with two types of answers; a stem and distractors. Similar to [20], we utilize aspects of Bloom's taxonomy to guide the generation of high-quality questions.

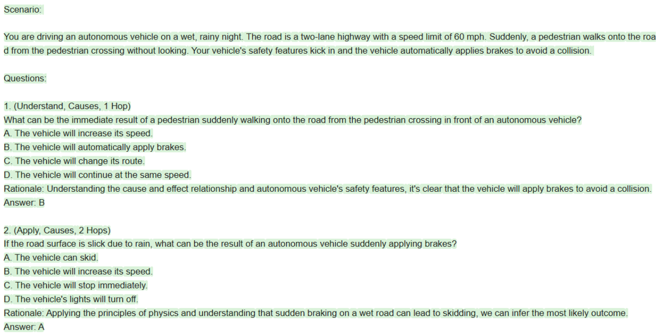
Information from all the previous subsections results in a comprehensive prompt to generate questions. Questions are to be generated in sets of variable numbers of questions. In the next section, we detail the experimental setup to evaluate this generation process. In addition, we evaluate pre-trained LLM's reasoning capabilities by providing them with a scenario and corresponding questions from the generated dataset and grading them based on their explanations and accuracy.

IV. Experimental Setup

In this section, we detail the test runs for evaluating the question-answer generation process and the performance of pre-trained LLMs in answering questions. The question-answer generation process uses GPT-4, an LLM. Since LLMs are probabilistic models with randomization, we need to evaluate several permutations of prompts to ensure that the generated question set is of high quality.

A. Question-Answer Generation

We carry out test runs to evaluate the performance of the QA generation process detailed in section III. Each test run consists of a configuration of the following parameters: (1) **Prompt Wording**. This includes the wording of specific aspects of the prompt detailed in section III. (2) **Ordering**. This pertains to the ordering of the different aspects of the prompt. (3) **Reasoning type**. (4) **Number of examples provided**. Following the Chain-of-thought prompting process in [26], this parameter details the number of examples. (5) **Elements removed from the base prompt**. Following the process of ablation testing, we measure the impact of parts of the prompt being removed. (6) **Number of questions generated**. (7) **Reasoning hops needed**. Following [27], we categorize reasoning-based questions in terms of reasoning hops needed to solve them. This provides us with a measure of the difficulty of a question. (8) **Number of iterations in a single chat context**. Since the LLM used is a *chat completion* model, the sequence of inputs and outputs have a message-based chat format. This, in addition to other advantages of these LLMs, means that we can refine outputs by iteratively generating the same response with some feedback [28]. We test whether this process improves the questions or not. (8) **Temperature**. LLMs contain a *temperature* parameter to introduce randomness and creativity in outputs. We evaluate the impact of this parameter, especially on the generation consistency. (9) **Pre-trained LLM used to generate questions**. (10) **Consistency in separate chat contexts**. Since the LLMs have randomness in their generation process, getting consistent outputs is difficult. We evaluate this. (11) **Solution and explanation generation**. We evaluate the generated questions with and without solution and explanation generation (and the impact of their ordering).



Scenario:

You are driving an autonomous vehicle on a wet, rainy night. The road is a two-lane highway with a speed limit of 60 mph. Suddenly, a pedestrian walks onto the road from the pedestrian crossing without looking. Your vehicle's safety features kick in and the vehicle automatically applies brakes to avoid a collision.

Questions:

1. (Understand, Causes, 1 Hop)
What can be the immediate result of a pedestrian suddenly walking onto the road from the pedestrian crossing in front of an autonomous vehicle?
A. The vehicle will increase its speed.
B. The vehicle will automatically apply brakes.
C. The vehicle will change its route.
D. The vehicle will continue at the same speed.
Rationale: Understanding the cause and effect relationship and autonomous vehicle's safety features, it's clear that the vehicle will apply brakes to avoid a collision.
Answer: B

2. (Apply, Causes, 2 Hops)
If the road surface is slick due to rain, what can be the result of an autonomous vehicle suddenly applying brakes?
A. The vehicle can skid.
B. The vehicle will increase its speed.
C. The vehicle will stop immediately.
D. The vehicle's lights will turn off.
Rationale: Applying the principles of physics and understanding that sudden braking on a wet road can lead to skidding, we can infer the most likely outcome.
Answer: A

Fig. 3: QA Generation

We carry out an adequate number of permutations of this configuration. We make use of the *guidance* python package [29]. We observe and record the questions generated by the LLM. An example of one such QA set generation is shown in figure 3. We evaluate for consistency in the questions (with possible answer choices) generated and the request made in the prompt. In addition, we make a statement regarding the plausibility of the solution and explanation generated. We measure the final number of questions generated, the redundancy among the questions, and the number of hops to reach the solution.

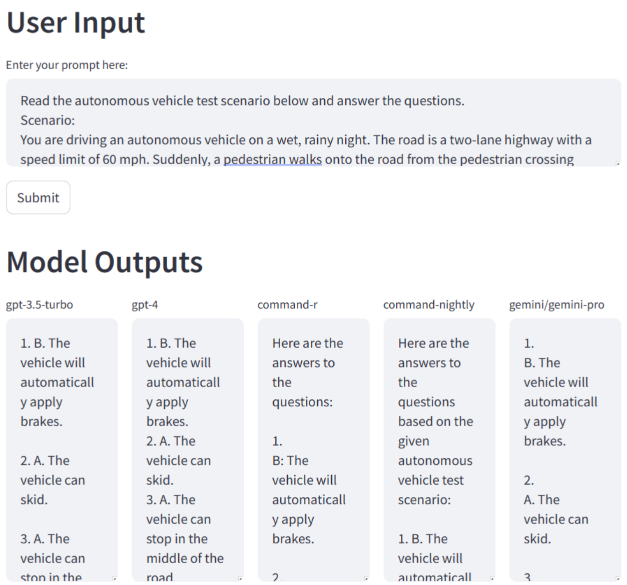
B. Grading Pre-trained LLMs

We carry out test runs to evaluate the performance of pre-trained LLMs given only a scenario and the corresponding set of questions. Each test run is a configuration of the parameters for the QA generation experiment above but applied to the questions. We manually grade the explanations for good reasoning skills. In addition, we record the accuracy of the options selected by pre-trained LLMs while attempting these questions, with a focus on LLMs *other than* the one that generated these questions. A summary of this evaluation is written in the next section. To increase the efficiency of the test runs, we make use of an *LLM playground* feature of the *LiteLLM* package that allows us to evaluate the response of multiple pre-trained LLMs simultaneously [30]. Examples of these are shown in figure 4.

V. Results and Discussion

In this section, we present summaries of evaluations for test runs under both experiment A and experiment B. Further, we analyze the results and detail the impact of each test run parameter. From the results of these test runs, we aim to address the following high-level research question:

- **RQ 1:** How do we generate effective questions using pre-trained LLMs to evaluate reasoning capabilities in the area of scenario-based testing of AVs?
- **RQ 2:** How do we effectively evaluate pre-trained LLMs using given QA data?



Enter your prompt here:

Read the autonomous vehicle test scenario below and answer the questions.
Scenario:
You are driving an autonomous vehicle on a wet, rainy night. The road is a two-lane highway with a speed limit of 60 mph. Suddenly, a pedestrian walks onto the road from the pedestrian crossing

gpt-3.5-turbo

1. B. The vehicle will automatically apply brakes.

2. A. The vehicle can skid.

3. A. The vehicle can stop in the

gpt-4

1. B. The vehicle will automatically apply brakes.
2. A. The vehicle can skid.
3. A. The vehicle can stop in the middle of the road.

command-r

Here are the answers to the questions:

1.
B: The vehicle will automatically apply brakes.

2.

command-nightly

Here are the answers to the questions based on the given autonomous vehicle test scenario:

1. B. The vehicle will automaticall

gemini/gemini-pro

1.
B. The vehicle will automatically apply brakes.

2.
A. The vehicle can skid.

3.

Fig. 4: Pre-trained LLM evaluations: LiteLLM Playground

A. *Question-Answer Generation Results*

We used the pre-trained LLM GPT-4 for evaluating the QA generation process. By carefully changing the experiment parameters detailed in the previous section, we gained valuable insights regarding effective prompting strategies for generating high-quality QA sets, correctness and consistency in generation, and managing redundant questions and answer choices. These insights are detailed below.

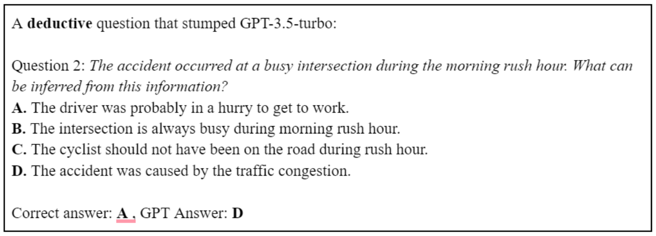
1) Effective prompting strategies: The elements within the generated questions or QA pairs were nearly always consistent with the instructions provided in the prompts. The scenarios generated were plausible and made use of the generated ontology. It must be noted though that multiple, simultaneous instructions about maintaining counts were difficult for the LLM to follow. For example, in one test run, we instructed the LLM to generate 10 QA pairs with a certain number from each Bloom's taxonomy level and a certain number of questions for 1, 2, and 3 reasoning hops. GPT-4 couldn't maintain all three simultaneous counts correctly in any of the test runs at any of the tested temperature levels. At the same time, the LLM was able to follow multiple, simultaneous instructions regarding *the structure of text generation*. In our prompt, we require GPT-4 to generate multiple choice questions with four options, follow Bloom's taxonomy, and follow the reasoning examples which it carried out quite well.

In line with the findings of [26], examples of reasoning were crucial for improving the generation process. We observed that removing examples of reasoning and simply instructing it to follow a reasoning pattern either reduced the questions requiring reasoning or removed them completely. The more the number and relevance of given examples, the better the generation. This process has been defined as *in-context learning*, which is a property that emerges as we scale from small to large language models [31]. In our experiments, at low-temperature values, we observed GPT-4 learning much subtler patterns within examples such as the answer choice which is supposed to be the correct answer and the writing style of answer choices such as the incorrect answer choices for commonsense reasoning were generally implausible. Since the temperature parameter introduces a certain degree of randomness in the next word selection, increasing this value generally reduced *overfitting* to the given examples. Along a similar line, we observed that adding *incorrect* examples e.g. questions with random words or characters for answer choices resulted in a significant reduction in QA writing quality and a significant increase in redundant questions and answer choices.

The presence of Bloom's taxonomy terms (adapted from [20]) significantly increased the *depth* of the QA pairs. When removed, the questions did follow the reasoning examples but only superficially, typically concerning the attributes of the scenario elements. Here, a significant insight would be that Bloom's taxonomy terms in addition to good examples with implicit patterns are crucial for high-quality QA generation. Regarding a few categories though, questions could often belong to multiple categories since a plausible rationale could be generated for each one.

Mentioning the number of reasoning hops required wasn't adequate in generating complex and challenging questions. However, when examples were provided, the LLM was able to generate complex questions. Increasing complexity of the questions though would often result in multiple correct options being plausible solutions or incorrect options being generated. A significant insight here would be that more examples of challenging questions with complex reasoning hops are needed in addition to a review process.

2) Managing redundancy: One significant need in this paper was to have reproducible results. This is challenging since LLMs are probabilistic models with significant randomness (with the presence of the *temperature parameter*). We observed redundancy in most cases in either the questions generated or the options provided for the questions. In some cases, since questions were on similar topics, some would provide facts that would answer a previous question. Increasing the temperature parameter and *specifying per question structure* would significantly reduce redundancy and make the QA set more diverse. However, with increased temperature, the same prompt in a new chat context would generate completely different scenarios and questions thereby not allowing for experiments to be reproduced at a concrete level. At an abstract level, this is still possible. At a zero temperature value, the generated scenarios and questions would have an increased consistency across new chat contexts but the relation between prompt and generated data still wouldn't be deterministic. Other features such as the OpenAI API's *reproducible outputs* provide the option to ensure increased determinism [32]. However, we weren't able to make use of this *seed* parameter within the *guidance* package that we were using.



A **deductive** question that stumped GPT-3.5-turbo:

Question 2: *The accident occurred at a busy intersection during the morning rush hour. What can be inferred from this information?*
A. The driver was probably in a hurry to get to work.
B. The intersection is always busy during morning rush hour.
C. The cyclist should not have been on the road during rush hour.
D. The accident was caused by the traffic congestion.

Correct answer: **A** , GPT Answer: **D**

Fig. 5: A question that stumped GPT-3.5-turbo

B. LLM Evaluation Results

We evaluated 4 pre-trained LLMs on the generated scenarios: OpenAI's *GPT-3.5* and *GPT-4*, Cohere's *Command-R* and *Command-R nightly* and Google's *Gemini Pro*. We made observations on the accuracy of the responses and the explanations generated. We observed the response of all LLMs to a scenario provided and the questions generated. The *LiteLLM* based experimental setup is shown in figure 4. All LLMs generally were able to reason well for the questions provided and provided accurate answer choice selections as well as explanations. The Cohere LLMs provided detailed explanations before selecting options. In select cases, as shown in figure 5, the LLM makes an error in answer choice selection.

Contrary to results in [15], in our evaluation, GPT-4 performs well in spatial reasoning and multiple-hop reasoning questions. There might be multiple explanations for this: (1) A significant time has passed since the evaluations of [15] and pre-trained LLMs have been updated since then. (2) The LLMs may reason differently for different concepts. Uniquely in this paper, reasoning evaluations are carried out in the context of AV test scenarios. More investigation is needed to verify this theory.

C. Threats to Validity

In this subsection, we detail any doubts that might threaten the validity of the results stated above. We then detail steps taken to mitigate these threats.

1) Circular Evaluation: One concern that may arise is that pre-trained LLMs are being used for QA generation and for evaluation as well. This may not produce valuable results since the training process is the same for both question generation and grading. We mitigate this threat in the following ways: (1) We ensure that the scenarios generated is close to the concepts in the prompt. We ensure that the concepts are an extension of the ASAM OpenXOntology. In addition, we check scenarios generated for plausibility. (2) We ensure that structure regarding answer choice and reasoning is introduced in the prompt so as to not rely on the LLM's knowledge. (3) We make use of GPT-4 for QA generation but also evaluate other models such as Cohere's Command-R and Google's Gemini-Pro. (4) We perform an evaluation on a new chat context and only share the scenario and the questions, not the information in the QA generation prompt.

VI. Conclusion and Future Work

Pre-trained LLMs present a significant potential for the domain of scenario-based testing of AVs. However, the reasoning capabilities of pre-trained LLMs within this context remain unknown. QA benchmarks are often used to assess the capabilities of pre-trained LLMs. In this paper, we propose an approach to create this benchmark and carry out evaluations with the help of LLMs themselves. Integrating approaches from MCQ generation, reasoning QA benchmarks, and ontology generation using LLMs, we can generate QA data for a wide range of scenarios and reasoning types. With the help of a comprehensive evaluation, we gain valuable insights regarding the topic of QA generation.

For future work, we can compare human-written and LLM-generated explanations for questions with BLEU and ROUGE metrics to compare the differences. In addition, more work can be carried out for generating *complex reasoning* questions, analysis of metrics such as *average length of questions* and *answer choice distribution*, and an evaluation of a larger variety of LLMs.

References

[1] J. Fleetwood, "Public health, ethics, and autonomous vehicles," *American Journal of Public Health*, vol. 107, no. 4, pp. 532–537, Apr. 2017, Epub 2017 Feb 16. DOI: 10.2105/AJPH.2016.303628.

[2] X. Wu, J. Cao, and F. Douma, "The impacts of vehicle automation on transport-disadvantaged people," *Transportation Research Interdisciplinary Perspectives*, vol. 11, p. 100 447, 2021, ISSN: 2590-1982. DOI: https://doi.org/10.1016/j.trip.2021.100447. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590198221001536.

[3] Ó. Silva, R. Cordera, E. González-González, and S. Nogués, "Environmental impacts of autonomous vehicles: A review of the scientific literature," *Science of The Total Environment*, vol. 830, p. 154 615, 2022, ISSN: 0048-9697. DOI: https://doi.org/10.1016/j.scitotenv.2022.154615. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0048969722017089.

[4] S. Feng, X. Yan, H. Sun, *et al.*, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Communications*, vol. 12, p. 748, 2021. DOI: 10.1038/s41467-021-21007-8. [Online]. Available: https://doi.org/10.1038/s41467-021-21007-8.

[5] Center for Sustainable Systems, University of Michigan, *Autonomous vehicles factsheet*, Pub. No. CSS16-18, 2023.

[6] [Online]. Available: https://www.nvidia.com/en-us/glossary/large-language-models/#:~:text=Large%20Language%20Models%20Explained,content%20using%20very%20large%20datasets..

[7] Z. Jiang, J. Araki, H. Ding, and G. Neubig, "How can we know when language models know? on the calibration of language models for question answering," eng, *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 962–977, 2021, ISSN: 2307-387X.

[8] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4149–4158. DOI: 10.18653/v1/N19-1421. [Online]. Available: https://aclanthology.org/N19-1421.

[9] B. Dalvi, P. Jansen, O. Tafjord, *et al.*, *Explaining answers with entailment trees*, 2022. arXiv: 2104.08661 [cs.CL].

[10] K. Sinha, S. Sodhani, J. Dong, J. Pineau, and W. L. Hamilton, *Clutrr: A diagnostic benchmark for inductive reasoning from text*, 2019. arXiv: 1908.06177 [cs.LG].

[11] Y. Bai and D. Z. Wang, "More than reading comprehension: A survey on datasets and metrics of textual question answering," *CoRR*, vol. abs/2109.12264, 2021. arXiv: 2109.12264. [Online]. Available: https://arxiv.org/abs/2109.12264.

[12] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1821–1827. DOI: 10.1109/IVS.2018.8500406.

[13] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87 456–87 477, 2020. DOI: 10.1109/ACCESS.2020.2993730.

[14] Y. Deng, J. Yao, Z. Tu, X. Zheng, M. Zhang, and T. Zhang, *Target: Automated scenario generation from traffic rules for testing autonomous vehicles*, 2023. arXiv: 2305.06018 [cs.SE].

[15] Y. Bang, S. Cahyawijaya, N. Lee, *et al.*, *A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity*, 2023. arXiv: 2302.04023 [cs.CL].

[16] P. Lu, S. Mishra, T. Xia, *et al.*, "Learn to explain: Multimodal reasoning via thought chains for science question answering," in *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[17] Z. Xu, Y. Zhang, E. Xie, *et al.*, *Drivegpt4: Interpretable end-to-end autonomous driving via large language model*, 2024. arXiv: 2310.01412 [cs.CV].

[18] J. Mao, J. Ye, Y. Qian, M. Pavone, and Y. Wang, *A language agent for autonomous driving*, 2023. arXiv: 2311.10813 [cs.CV].

[19] Y. Tang, A. A. B. da Costa, J. Zhang, I. Patrick, S. Khastgir, and P. Jennings, *Domain knowledge distillation from large language model: An empirical study in the autonomous driving domain*, 2023. arXiv: 2307.11769 [cs.CL].

[20] J. Doughty, Z. Wan, A. Bompelli, *et al.*, "A comparative study of ai-generated (gpt-4) and human-crafted mcqs in programming education," in *Proceedings of the 26th Australasian Computing Education Conference*, ser. ACE 2024, ACM, Jan. 2024. DOI: 10.1145/3636243.3636256. [Online]. Available: http://dx.doi.org/10.1145/3636243.3636256.

[21] J. Weston, A. Bordes, S. Chopra, *et al.*, "Towards ai-complete question answering: A set of prerequisite toy tasks," *arXiv preprint arXiv:1502.05698*, 2015.

[22] R. Mirzaee, H. R. Faghihi, Q. Ning, and P. Kordjmashidi, *Spartqa: : A textual question answering benchmark for spatial reasoning*, 2021. arXiv: 2104.05832 [cs.CL].

[23] L. Qin, A. Gupta, S. Upadhyay, L. He, Y. Choi, and M. Faruqui, *Timedial: Temporal commonsense reasoning in dialog*, 2021. arXiv: 2106.04571 [cs.CL].

[24] ASAM, *Asam openxontology*, https://www.asam.net/project-detail/asam-openxontology/, Web page, accessed 22 April 2024, 2024.

[25] M. Scholtes, L. Westhofen, L. R. Turner, *et al.*, "6-layer model for a structured description and categorization of urban traffic and environment," *IEEE Access*, vol. 9, pp. 59 131–59 147, 2021.

[26] J. Wei, X. Wang, D. Schuurmans, *et al.*, *Chain-of-thought prompting elicits reasoning in large language models*, 2023. arXiv: 2201.11903 [cs.CL].

[27] T. Qian, J. Chen, L. Zhuo, Y. Jiao, and Y.-G. Jiang, "Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 4542–4550.

[28] A. Madaan, N. Tandon, P. Gupta, *et al.*, "Self-refine: Iterative refinement with self-feedback," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[29] S. Lundberg *et al.*, *Guidance*, https://github.com/guidance-ai/guidance, Web page, accessed May 01, 2024, 2024.

[30] I. Jaff *et al.*, *Litellm*, https://github.com/BerriAI/litellm, Web page, accessed May 01, 2024, 2024.

[31] J. Wei, Y. Tay, R. Bommasani, *et al.*, "Emergent abilities of large language models," *arXiv preprint arXiv:2206.07682*, 2022.

[32] S. Anadkat, *How to make your completions outputs consistent with the new seed parameter*, https://cookbook.openai.com/examples/reproducible_outputs_with_the_seed_parameter, Nov. 2023.